ARY
SCHOOL

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (*If applicable*) 37	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000	7b. ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (*If applicable*)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (*City, State, and ZIP Code*)	10. SOURCE OF FUNDING NUMBERS Program Element No / Project No / Task No / Work Unit Accession Number

11. TITLE (*Include Security Classification*)
NAVAL POSTGRADUATE SCHOOL SCHEDULING SYSTEM (NPS^3)

12. PERSONAL AUTHOR(S) Dowler, Richard C.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (*year, month, day*)	15. PAGE COUNT
Master's Thesis	From To	September 1992	301

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS (*continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUBGROUP	Schedules; Scheduling; Sysstems Analysis and Design; Structured Analysis; Structured Design

19. ABSTRACT (*continue on reverse if necessary and identify by block number*)

A database management system, the Naval Postgraduate School Scheduling System (NPS^3), is proposed to support the Naval Postgraduate School schedulers in scheduling academic events and courses for students, instructors and classrooms during an academic quarter. NPS^3 is designed to assist schedulers in the scheduling process by presenting courses to be scheduled based on scheduling priorities. Semi-permanent events are automatically scheduled and entered in relevant student, instructor and classroom schedules based on predetermined entries in the NPS^3 database. Schedulers are allowed to assign time periods and classrooms to each subsequent course. NPS^3 then enters the scheduling information in the appropriate student, instructor and room schedules before the next course is scheduled. This process continues until the lowest priority course is scheduled. Structured Analysis and Design Methods are used to present NPS^3. High-level design specifications are detailed with data flow diagrams, entity relationship diagrams and structure charts.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Daniel R. Dolk	22b. TELEPHONE (*Include Area code*) (408) 646-2260	22c. OFFICE SYMBOL Code AS/Dk

T259707

Approved for public release; distribution is unlimited.

Naval Postgraduate School Scheduling System (NPS^3)

by

Richard C. Dowler
Captain, United States Marine Corps
B.S., University of Missouri-Columbia, 1986

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1992

ABSTRACT

A database management system, the Naval Postgraduate School Scheduling System (NPS^3), is proposed to support the Naval Postgraduate School schedulers in scheduling academic events and courses for students, instructors and classrooms during an academic quarter. NPS^3 is designed to assist schedulers in the scheduling process by presenting courses to be scheduled based on scheduling priorities. Semi-permanent events are automatically scheduled and entered in relevant student, instructor and room schedules based on predetermined entries in the NPS^3 database. Schedulers are allowed to assign time periods and classrooms to each subsequent course. NPS^3 then enters the scheduling information in the appropriate student, instructor and room schedules before the next course is scheduled. This process continues until the lowest priority course is scheduled. Structured Systems Analysis and Design Methods are used to present NPS^3. High-level design specifications are detailed with data flow diagrams, entity relationship diagrams and structure charts.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. PROBLEM 2

B. METHODOLOGY 3

C. STRUCTURE OF THIS THESIS 3

II. BACKGROUND ON NPS SCHEDULING 5

A. NAVAL POSTGRADUATE SCHOOL (NPS) 5

B. CURRENT NPS COURSE SCHEDULING 6

1. Forecasting 7

2. Pre-Scheduling 7

3. Scheduling 7

4. Post-Scheduling 8

5. Synopsis of the Present Situation 8

6. Most Pressing Concerns 9

C. NAVAL POSTGRADUATE SCHOOL SCHEDULING SYSTEM (NPS^3) 10

1. Function 10

2. Extending NPS^4 11

3. Concurrent Access of Data 13

III. STRUCTURED METHODOLOGY 15

A. DATA FLOW DIAGRAMS 15

B. ENTITY RELATIONSHIP DIAGRAMS 17
C. STRUCTURED DESIGN USING STRUCTURE CHARTS . . . 20
D. MODULE SPECIFICATIONS 22

IV. CONCLUSION/RECOMMENDATION 24
A. WHY STRUCTURED METHODOLOGY? 24
B. IMPLEMENTATION 25
C. SOFTWARE 26
D. OMNIS 7 28

APPENDIX A: CURRENT NPS^3 PHYSICAL DATA FLOW DIAGRAMS . 31

APPENDIX B: NPS^3 ENTITY RELATIONSHIP DIAGRAMS 38

APPENDIX C: NPS^3 LOGICAL DATA FLOW DIAGRAMS 71

APPENDIX D: NPS^3 STRUCTURE CHARTS 110

APPENDIX E: MODULE SPECIFICATIONS 155

LIST OF REFERENCES 291

INITIAL DISTIBUTION LIST 293

I. INTRODUCTION

In March of 1992, two Naval Postgraduate School (NPS) students, Nolan J.S. and Youngblood P.D., proposed a decision support system (DSS), "NAVAL POSTGRADUATE SCHOOL SCHEDULING SUPPORT SYSTEM (NPS^4)", to assist the NPS course schedulers in developing student, instructor, room and final exam schedules for an academic quarter. (Nolan, 1992)

Their thesis research consisted of an extensive systems analysis which included an historical study of the evolution of the NPS scheduling process and an examination of every detail of the current scheduling system. They provided a definition of scheduling needs and requirements, some database specifications, proposals and recommendations to develop a DSS and the evaluation of alternative solutions. Additionally, a user interface prototype was presented. (Nolan, 1992)

In NPS^4, Nolan and Youngblood presented a number of alternative solutions for the NPS scheduling process. An analysis of their research reveals the following recommendations common to the majority of the alternative solutions:

- augmentation of parts of the existing system is presumably the most prudent option in terms of maintaining the human element in the scheduling process, easing user transition into a new system and holding down implementation costs;

- any automated portion of the current scheduling system should have the capability to accommodate at least two course schedulers, offering concurrent access and use of the scheduling data and partially completed schedules; and
- a dedicated relational database is needed in almost all alternative solutions to store the huge amount of important information (much of which resides on the NPS mainframe computer) required to construct student course group, instructor and room schedules. (Nolan, 1992)

A. PROBLEM

As previously mentioned, Nolan and Youngblood (1992) presented a user interface prototype to validate the course scheduler's requirements. This was accomplished using a rapid prototyping methodology as a tool for software development. However, there is a need for multiple methodologies in developing large systems such as that presented by the NPS scheduling process (Keyes, 1992). In today's changing world of software development, structured analysis and design methods can be integrated with prototyping techniques (Keuffel, March 1992), "...coming together in a combination that is better than any of the parts...." (Keyes, 1992) in order to shorten the system development life cycle.

The purpose and goal of this thesis research is to provide a more detailed systems analysis and design specifications of the current NPS scheduling process and the target system presented and proposed respectively by Nolan and Youngblood in NPS^4. We envision that these specifications will be used by computer programmers for eventual systems implementation of a

computer based solution/enhancement to the current scheduling system or to NPS[4]. Structured Systems Analysis and Design methodologies will be used with emphasis on database design.

B. METHODOLOGY

Semi-formal analysis and design specifications will be developed using dataflow diagrams, structure charts and module specifications, complemented by database specifications consisting of entity relationship diagrams and entity and attribute definitions.

C. STRUCTURE OF THIS THESIS

Chapter II provides a brief discussion of the organization of the Naval Postgraduate School and the background of the current NPS course scheduling process.

Chapter III presents a concise overview of dataflow diagrams, entity relationship diagrams, structure charts and module specifications.

Chapter IV summarizes the conclusions and recommendations for follow-on steps required to implement the design generated in this thesis.

Appendices A, B, C, D and E contain the physical dataflow diagrams of the current NPS scheduling system, the entity relationship diagrams with related entity and attribute definitions, the logical dataflow diagrams of the target scheduling system, the target system structure charts, and the module specifications, respectively.

II. BACKGROUND ON NPS SCHEDULING

A. NAVAL POSTGRADUATE SCHOOL (NPS)

The Naval Postgraduate School is atypical compared to most public and private academic institutions with respect to its course scheduling process. For the sake of clarity, a brief description of the NPS scheduling process summarized from Nolan and Youngblood (1992) is presented below:

Under the auspices of the Chief of Naval Operations, the Naval Postgraduate School located at Monterey, California, is an academic institution of higher learning and research with the primary purpose to increase the combat effectiveness of U.S. and allied armed forces by providing graduate degree programs to qualified students in areas not usually available through other educational institutions, and by conducting research that actively supports U.S. armed forces operations. (Nolan, 1992)

The student body at NPS is composed of U.S. military officers from all branches of the service, foreign national officers from almost 30 allied countries, and some U.S. military enlisted personnel and federal civilian employees. Most study and research programs pursued satisfy both Masters or Doctorate degree requirements and fulfil requirements that

earn a military occupational specialty (MOS) code for service members. (Nolan, 1992)

The NPS administration of learning and research is a cooperative effort between military and civilian personnel. Academic departments, with a majority of their faculty members being civilians with Doctorate degrees, prepare and offer courses. Curricular programs of study are developed by military Curricular Officers, who direct students through chosen programs to ensure students complete both degree and MOS requirements. Civilian Academic Associates work with Curricular Officers to ensure that each curricular program is academically sound. A civilian Management Analyst manages the collection of Departmental and Curricular Officer data used in the scheduling process, and the NPS Class Schedulers construct course schedules from this collected data. (Nolan, 1992)

B. CURRENT NPS COURSE SCHEDULING

The academic year at NPS is divided into four quarters, Fall, Winter, Spring and Summer. The scheduling process consists of four phases and requires a joint effort from several groups of people in order to construct schedules during each quarter. An excerpt from NPS[4] (Nolan, 1992) summarizes the four phases of the scheduling process at NPS:

1. Forecasting

Up to a year or more in advance, departmental schedulers and the Management Analyst forecast probable quarterly student course demands and Instructor workloads necessary to meet these demands. A Tentative Course Schedule is produced indicating which courses are expected to be taught by each academic department during each quarter. (Nolan, 1992)

2. Pre-Scheduling

Pre-Scheduling is a quarterly process. Curricular Officers, via the Management Analyst, give departmental schedulers information about which courses their students are requesting for that quarter. Departments then determine which courses will actually be taught that quarter. If there are requested courses that will not be taught, students replace these with ones that will be. After receiving this updated information, departments divide courses that are too large into segments and assign instructors to teach each course. (Nolan, 1992)

3. Scheduling

Pre-Scheduling information is given to the Class Schedulers, who use it to construct a Master Instruction Schedule and individual schedules for students, instructors

and classrooms. Schedules for the upcoming quarter are then distributed to applicable offices and individuals. (Nolan, 1992)

4. Post-Scheduling

After all the schedules are constructed including the Master Instruction Schedule, and throughout the scheduled quarter, changes may be made to the student, Instructor or room schedules. Student schedule changes are approved by their Curricular Officer and Academic Associate. Instructor schedule changes to a course's room or time periods are coordinated by the Class Schedulers and the changes are registered and maintained by them. (Nolan, 1992)

5. Synopsis of the Present Situation

Each quarter during the scheduling phase, course scheduling is done entirely by manual manipulation of information from Department Chairmen and Curricular Officers by two Class Schedulers. The Schedulers, with pencil and 5" x 8" schedule cards use "time-proven heuristics," (Nolan, 1992) to assign time periods and rooms literally by hand to hundreds of courses and course segments offered during a quarter. These courses and course segments are then matched to approximately 300 Instructor schedules and 1100 student course group schedules. Additionally each quarter, room schedules for each academic room used for instruction at NPS, are manually generated. (Nolan, 1992)

Course Schedulers use a set of scheduling priority guidelines, as well as Instructor preferences taken from Department Chairman Reports, to help minimize scheduling conflicts. However, when conflicts do occur, such as time period and/or room conflicts, or changes in course scheduling are requested, each conflict or change request is manually resolved, subsequently changing each schedule card affected. (Nolan, 1992)

In addition to the original schedule cards that are constructed each quarter, each original schedule is photocopied multiple times and distributed to their respective recipients.

After regular course scheduling is completed, final exam schedules must be constructed for each course that requires a final exam, assigning rooms and time periods to each final exam. As with regular course scheduling, conflicts must be minimized, resolved and avoided.

The present scheduling process at the time of this thesis research is seen as being vastly inefficient in regard to spending valuable man-hours making tedious manual entries and erasures to schedules and their related schedules, and photocopying thousands of schedule cards for distribution.

6. Most Pressing Concerns

Thesis research done by Nolan and Youngblood (1992) in "The Naval Postgraduate School Scheduling Support System,"

identified the most pressing concerns regarding the current NPS scheduling process:

- "...constantly having to locate, retrieve, and replace physical schedule cards...." (Nolan 1992) related to each scheduled course and course segment in order to add, modify or delete schedule information;
- "...lack of standardization in department entries to the [final] iteration of the Department Chairman Report...." (Nolan, 1992) indicating whether time periods and rooms are required or preferred by instructors; and
- lack of proper documentation methodology for changes to schedules during the Post-Scheduling phase. (Nolan, 1992)

In their analysis, Nolan and Youngblood, emphasized the need to maintain the human element required in the complex scheduling construction at NPS, and allowing the continuation of the freedoms of choice enjoyed in the present scheduling process. Since maintaining the human element is paramount in any given solution to meet the scheduling system requirements at NPS, Nolan and Youngblood proposed a partly automated decision support system, extending the present scheduling system capacity and creating more system capabilities. (Nolan, 1992)

C. NAVAL POSTGRADUATE SCHOOL SCHEDULING SYSTEM (NPS^3)

1. Function

A database management system, the Naval Postgraduate School Scheduling System (NPS^3) is proposed to support the NPS course schedulers in scheduling academic events and courses for students, instructors and classrooms during an academic

quarter. NPS^3 is designed to assist schedulers in the scheduling process by presenting courses to be scheduled based on scheduling priorities (Nolan, 1992). Semi-permanent events are automatically scheduled and entered in relevant student, instructor and room schedules based on predetermined entries in the NPS^3 database. Schedulers are allowed to assign time periods and classrooms to each subsequent course, course segment or laboratory. After resolving any scheduling conflicts, NPS^3 then enters the scheduling information in the appropriate student, instructor and room schedules before the next course, course segment or laboratory is scheduled. This process continues until the lowest priority course is scheduled. (Interview, Schedulers, April 13, 1992)

2. Extending NPS^4

Nolan and Youngblood (1992) used a rapid prototyping strategy during the analysis phase of system development demonstrating a user interface prototype "...to validate the analyst's view of what the user's want and need...." (Keuffel, October 1991). This user interface prototype was developed "...using Hypercard 2.0 on an Apple Macintosh IIci personal computer with 8MB of RAM and a 64K cache card under the System 7 operating system." (Nolan, 1992)

There are limitations to using rapid prototyping in systems development: it is "...hard to control, and has no overall plan, intermediate deliverables, 'audit trail,' or certainty about the result". (Gardner, 1991)

Furthermore, using Apple Macintosh's Hypercard environment to develop a scheduling system for the NPS course schedulers presents a dilemma. Currently, NPS course schedulers operate in an IBM compatible PC environment, making the use of Hypercard for continuation in system development questionable.

Given the limitations of rapid prototyping, the incompatibility dilemma that Hypercard presents, and the enormity of the NPS scheduling process, a need exists for structured analysis and design documentation of the Naval Postgraduate School Scheduling System (NPS^3).

Structured methodologies provide "...guidance in completing complex tasks as well as providing a medium for consistent communication...." (Gardner, 1991) between analysts, designers and programmers. "Methodologies are based on theory and have been tested many times, so that they've acquired formal rigor." (Gardner, 1991) In addition, structured methodology offers an historical record of the system under development (Keuffel, March 1992).

NPS^3 is not intended to replace the work done by Nolan and Youngblood (1992), but to be used to complement their work in an integrated and interactive approach to the system

development. NPS^3 will facilitate communication between system development team members by partitioning the scheduling process "...into manageable chunks...." (Plauger, 1992). The scheduling process problem is "...decomposed into parts...." showing the "...modularity of design...." presenting "...a logical design before one tries the physical design." (Gardner, 1991). NPS^3 will answer questions such as:

- Where is the development process now?,
- What development has been accomplished to date?, and
- Where is the development process going next?

Moreover, addressing the Hypercard dilemma, NPS^3 structured analysis and design specifications will be developed "...to achieve hardware and software independence...." (Wetherbe, 1984). In doing so, NPS^3 allows future system implementors the freedom to choose the appropriate hardware and software for system implementation.

3. Concurrent Access of Data

An issue not addressed in the NPS^3 design specifications, but certainly important in developing a multiuser database scheduling system, is the need for concurrent access and use of scheduling data and partially completed schedules.

When the situation arises where two or more schedulers are using the same scheduling data and/or partially completed

schedules (e.g., two different schedulers are scheduling two different courses to the same instructor schedule), a certain degree of data and record locking is needed while at the same time allowing for the 'refreshing ' of the concurrent scheduler's screen as one scheduler manipulates data shared by both schedulers.

Care must be taken to ensure that there is not too little data or record locking which may cause integrity problems, nor too much data or record locking which may result in delays or deadlock. (PC User, June 17, 1992) Furthermore, the time taken to 'refresh' a scheduler's screen is a critical factor in the efficiency of the scheduling system. If not done properly, one scheduler could effectively update/erase an other scheduler's recent changes to the concurrent data and/or schedule.

To account for 'refreshing' a scheduler's screen, NPS[3] structure charts would have to be modified. The most logical place to 'refresh' a scheduler's screen is during the procedure for resolving schedule record conflicts, Appendix D, Figure D-22. When the system determines schedule conflicts, searches for alternative schedule selections, or puts alternative selection to the schedule record, multiuser screens accessing concurrent data and/or schedules can be checked and 'refreshed' with the relevant scheduling data where appropriate.

III. STRUCTURED METHODOLOGY

Structured methods provide an effective "...communication vehicle...." (Keyes, 1992) for analysts to capture the system requirements from its end users (Keuffel, March 1992) and provide traceability and maintainability to programmers during system design and implementation (Riehle, 1991). The structured methodologies used in this thesis are structured analysis, entity relationship diagrams and structured design. The structured analysis includes physical and logical data flow diagrams. Entity and attribute definitions are included with the entity relationship diagrams. The structured design includes structure charts and module specifications.

A. DATA FLOW DIAGRAMS

Dataflow diagrams (DFD) are models used widely in structured analysis. They are not flow charts and thus do not explicitly show the flow of control through a system. However, dataflow diagrams do show flow of data, storage of data, and the processes that respond to and change data. (Whitten, 1989)

Different language and symbol sets have been developed to illustrate dataflow diagrams. For the purposes of this thesis, and illustrated in Figure 3.1, the Demarco-Yourdon

symbol set, "...perhaps the most well known...." (Keyes, 1992), will be used where:

- A circle represents a process that transforms inputs into outputs. The details of the process are not known.
- A rectangle depicts a source/destination name that defines the boundaries of the system. Source/destinations provide the net inputs to and receive the net outputs from the system.
- Data stores are depicted as open-ended boxes.
- Named arrows are depicted as data flows, representing inputs or outputs.
- A slash in the lower right-hand corner of a source/destination indicates a repetition of a source/destination within the given data flow diagram.
- A double line to the left of a data store indicates that a data store is repeated within the given data flow diagram. (Whitten, 1989)

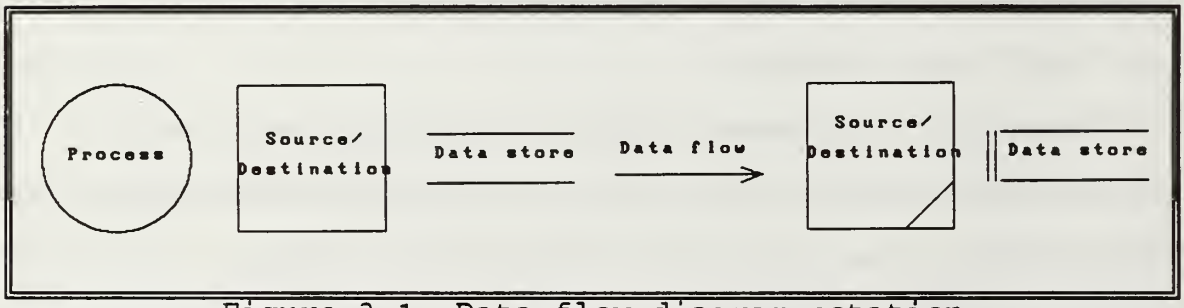


Figure 3.1 Data flow diagram notation

Several sets of dataflow diagrams are produced during structured analysis and include DFD's that:

- Model the current system and the target system to be built.

- Model the physical implementation details of the system, called the physical system, or model the essential characteristics of the system, called the logical system. (Whitten, 1989)

In Appendix A, the physical context data flow diagrams and the physical systems level data flow diagrams for the current NPS scheduling process are presented to detail the current scheduling system implementation.

In an attempt to gain a more detailed description of the NPS scheduling process user requirements presented by Nolan and Youngblood (1992), and to enhance possible alternative implementation solutions, the logical data flow diagrams are displayed in Appendix C. The logical data flow diagrams model the essential processing requirements of the system independent of any technology that might be used to implement those requirements (Whitten, 1989).

B. ENTITY RELATIONSHIP DIAGRAMS

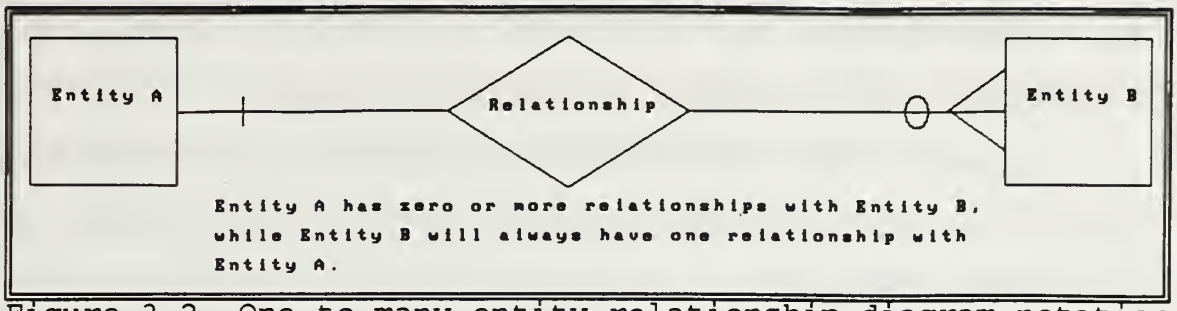
An Entity Relationship Diagram (ERD) is a logical systems analysis technique of data modeling that illustrates "...data at rest...." (Whitten, 1989). An ERD is a very detailed picture of data storage independent of the processing performed with those data stores. Therefore ERD's do not depict flow or processing of data and should not be read like data flow diagrams or flow charts. Because entity relationship diagrams depict data at rest or data being

stored, they do not imply how data is implemented, created, modified, used or deleted. (Whitten, 1989)

An ERD is made up of entities and the relationships between those entities. An entity is anything real or abstract that can be described by a set of common characteristics that are called data elements or attributes. Usually, at least one of the entity's data elements uniquely identifies one and only one occurrence of the entity. This data element or elements is referred to as the key. (Whitten, 1989)

A relationship is a natural association between one or more entities. Relationships usually exist between two different entities, but may also exist between different occurrences of the same entity. Furthermore, relationships may be dependent on other relationships. (Whitten, 1989)

As with data flow diagrams, various language and symbology is used to illustrate entity relationship diagrams, see Figure 3.2. Normally a rectangle represents an entity while a diamond represents a relationship. Because of the complexity of the NPS scheduling process and its many entities and relationships, the diamond is omitted to save space in the NPS Scheduling System ERD. (Whitten, 1989)


Entity A
Relationship
Entity B
Entity A has zero or more relationships with Entity B, while Entity B will always have one relationship with Entity A.
Figure 3.2 One-to-many entity relationship diagram notation

The NPS Scheduling System entity relationships are described by words and symbols that indicate the number of occurrences of one entity that can exist for a single occurrence of the related entity, and vice versa. Three general possibilities of occurrences exist:

- One-to-one (1:1)--for one occurrence of the first entity there can exist only one related occurrence of the second entity and vice versa;
- One-to-many (1:M or M:1)--for one occurrence of one entity there can exist many related occurrences of a second entity; it doesn't matter which is first or second;
- Many-to-many (M:M)--for one occurrence of the first entity there exists many related occurrences of the second entity, and for one occurrence of the second entity there exists many occurrences of the first entity. Many-to many relationships are transformed into one-to-many relationships creating an intersection between the many-to-many relationship. (Whitten, 1989)

Furthermore, the hash marks on the line between a relationship represent a mandatory relationship while a circle represents an optional relationship (Kroenke, 1988). A "...crow's foot...." (Haas, 1991) on a line between a relationship represents many occurrences of that relationship

while the absence of a "...crow's foot...." (Haas, 1991) represents one occurrence of that relationship.

Appendix B presents the Entity Relationship Diagrams and the entity and attribute definitions for the proposed NPS Scheduling System.

C. STRUCTURED DESIGN USING STRUCTURE CHARTS

Structured design is considered a disciplined approach to computer design that:

- develops "...a blueprint of a computer system solution to a problem that has the same components and interrelationships among the components as the original problem....
- ...seeks to conquer the complexity of large systems by means of partitioning the system into 'black boxes', and by organizing the black boxes into hierarchies suitable for computer implementation....
- [and] ...uses tools, especially graphic ones, to render systems readily understandable." (Page-Jones, 1988)

In Structured Design, structure charts are used to illustrate the partitioning of a system into modules, or 'black boxes', and show their hierarchy, organization and communication. The concern is almost entirely with what a module does rather than how it does it. This form of documentation actually helps the designers during the design effort. (Page-Jones, 1988) Structure charts present a "...template for coding....a plan of attack...." (Keuffel, October 1991), bridging the gap between the analysis and the

design of a system. Furthermore, structure charts are flexible enough to remain current with modification to the system over its lifetime. (Page-Jones, 1988)

In addition, these structure charts serve as a 'road map' for maintainers to use to quickly track down defects or implement user modifications. (Page-Jones, 1988)

Figure 3.3 illustrates structure chart notation. Modules are shown as rectangular boxes, with their names inside. The name is a statement of the module's function, what it does to completion each time it is called. Pre-defined modules are graphically illustrated by adding lines parallel to its vertical sides. Pre-defined modules already exist in a system or application library, operating system or database management system. Communication from one module to another, referred to as a 'call', is indicated by an arrow. Data that flows between modules is illustrated by using a circle with an arrow attached. (Page-Jones, 1988)

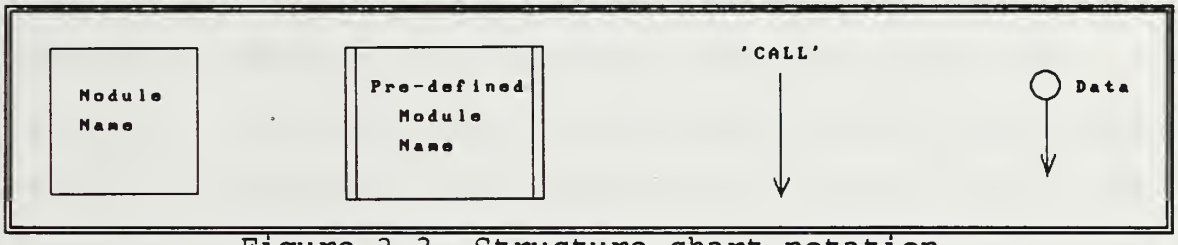


Figure 3.3 Structure chart notation

By using the scheduling system requirements detailed by Nolan and Youngblood (1992), and the specifications from Appendix C, Appendix D presents the NPS^3 structure charts illustrating the use of a database management system:

- to automatically import data from files downloaded from the NPS mainframe computer,
- to allow the user to update existing NPS^3 database files,
- to create NPS schedules by allowing users to schedule courses based on scheduling priorities, and
- to review and update existing schedule files.

D. MODULE SPECIFICATIONS

Like structure charts, module specifications are used as a communication tool to bridge the gap between design and programming, giving a potential programmer information on the procedural details of each module. The programmer is told what inputs the module uses when called, what outputs the module returns and the function the module is expected to carry out. (Page-Jones, 1988)

Two types of specifications are offered: interface specifications and specification by pseudocode. Interface specifications use structured English, providing "...a minimal amount of procedural detail to accompany the structure chart." (Page-Jones, 1988) This allows the programmer the freedom to program in what ever way he/she pleases.

Specification by pseudocode is a more detailed description of a module's internal procedure. Pseudocode simulates actual code and therefore presents less work for the programmer to attain the final code. (Page-Jones, 1988)

Appendix E provides the module specifications for the NPS^3 modules illustrated in Appendix D.

IV. CONCLUSION/RECOMMENDATION

A. WHY STRUCTURED METHODOLOGY?

The Naval Postgraduate School Scheduling System (NPS^3) detailed in this thesis does not pretend to offer a fool proof analysis and design solution to the NPS course scheduling process. However, NPS^3 is presented to reinforce and complement the work done by Nolan and Youngblood (1992) by suggesting a structured rational design approach.

A structured rational design approach to system development will offer understanding and guidance to potential programmers, facilitate control of the project rather than proceed in an adhoc manner, present a standard procedure for system implementation, make measuring the progress of a system easier and enhance good overall management of the system development. (Keuffel, September 1991)

To expedite the growth of the scheduling system, NPS^3 structured analysis and design specifications and Nolan and Youngblood's (1992) user interface prototype "...can be integrated into a new development life cycle that combines the benefits of both approaches...." (Plauger, 1992).

> "From structured methods, we gain the abstract representation of the system and the documentation that provides the historical record of the project. From rapid prototyping we gain the ability to quickly put before our users a concrete representation of user interfaces, which

iteratively evolves into the completed system." (Keuffel, March 1992)

B. IMPLEMENTATION

A system with the size and complexity of the Naval Postgraduate School Scheduling System can be implemented in a number of different ways.

The first step that should be taken for implementation is to build the relational database management system given the specifications in Appendix B. Once the relational database is built, integration with Nolan and Youngblood's user interface prototype should be investigated to incorporate the relational database and the user interface prototype into a "...functional prototype...." (The Computer Conference Analysis Newsletter, April 15, 1992). This would entail actually building part of the scheduling system. It is important to keep in mind that the NPS course schedulers should be included early and continually in the implementation and testing of the scheduling system.

The next step of implementation would be to divide the design presented in this thesis into logical components to be developed incrementally. Possible divisions of the system are:

- a component to read files downloaded from the NPS mainframe and import the data into the NPS^3 database files;

- a component that automatically schedules semi-permanent events to the appropriate schedules;
- a component that allows scheduling of courses based on a particular course scheduling priority, after which subsequent priorities could easily be implemented;
- a component that determines and resolves any scheduling conflicts that may occur;
- a component that 'refreshes' multiuser screens when concurrent access and use of scheduling data and partially completed schedules is taking place; and
- although not addressed in this thesis except to show their organization in the overall system, components for the Final Exam Scheduling, Master Instruction Schedule and Scheduling Data Graph applications, respectively.

C. SOFTWARE

There are a number of PC-based multiuser relational database management systems (RDBMS) on the market today that would be possible candidates for use in NPS[3]:

- Borland International Inc.'s dBase IV 1.5
- Borland International Inc.'s Paradox 4.0;
- Fox Software's FoxPro LAN 2.0;
- DataEase International Inc.'s DataEase 4.5;
- Microrim Inc.'s R:BASE 3.1c;
- Oracle Corp.'s Oracle 7.0; and
- Blyth Software Inc.'s Omnis 7.

Each of the RDBMS's listed above has its strengths and weaknesses relative to the other software packages. Although this is valuable information when presenting alternatives, to give a detailed analysis of the strengths and weaknesses of

each RDBMS would be a thesis in itself. Instead, the following major considerations are provided for selecting a multiuser RDBMS:

- Ready-to-use-tools--can a non-programmer produce adhoc queries and reports, create and maintain tables and indexes, and enter and edit data without custom coding?;
- Database maintenance--can the data be stored and accessed independent of any application?;
- Environment--does the RDBMS offer the development environment with the capability to produce the application needed to use the RDBMS in an effective manner?;
- Database structure--significant restrictions on the number of records or fields or the size of records or fields should be evaluated; are null values represented correctly and can they be tested for in the database?; is the data stored in one large repository which would cause efficiency problems or are the tables stored as separate files which would speed data access;
- Referential integrity--does the RDBMS automatically prevent erroneous data entries to records that don't exist or prevent deletion of an entity record without deleting it from its related entities; or does this function have to be programmed in by the user?;
- Data security and access control--what degree of security and access control is offered?; is it at the level needed?; will you have to write code to get it to the level you desire?;
- Record locking--this was addressed in Chapter II under Concurrent Access of Data;
- Data loading--can the RDBMS import data from other files or databases?; What file formats can it read and how many?; is the data validated before or after loading?;
- Structured Query Language supportability--does the RDBMS support embedded SQL for executing non-procedural operations on the data for ease-of-access, scaleability and optimization;

- Programming--how easy is it to customize the code to meet the requirements of specific applications? (PC User, June 17, 1992); and

- Hypercard interface--the RDBMS must have the capability to either provide a graphical user interface (GUI) as proposed by Nolan and Youngblood (1992), or else be able to connect to the Hypercard GUI developed in NPS[4].

D. OMNIS 7

In the author's view, Blyth Software Inc.'s Omnis 7 is a particularly desirable candidate for a relational database management system to be used in NPS[3].

Although relatively high priced ($1,250), Blyth Software's Omnis 7 offers a powerful relational database management system for single users or for multiusers of complex networked database systems. Specifically marketed for developers, Omnis 7 is easy-to-use and robust providing abundant features for beginners such as quick prototyping, a short popup menu mode that "...generates entry screens, reports, and menus at the click of a mouse...." (Miley, 1992), and "...online context sensitive help...." (Fogel, 1992).

Written in a high level procedural language, C++, Omnis 7 lets the developer/user take care of several discrete steps by issuing a single command. In short, Omnis 7 offers "...a sophisticated integrated development environment for creating database applications...." (Anderson, 1992).

One of the features that make Omnis 7 a true integrated development environment is its cross-platform architecture. Omnis 7

> "...offers seamless portability between Mac database applications created for use in the Mac environment and applications meant for one of the competing PC-based graphical user environments, Microsoft Windows or IBM's Presentation Manager. Applications written for Omnis Seven on any of these platforms can be moved to the other without modification." (Miley, 1992)

This cross-platform portability makes Omnis 7 a perfect candidate to be integrated with Nolan and Youngblood's (1992) user interface prototype. Alternatively, Omnis 7 can be developed as a file server based data manager in a LAN environment or "...as a front end to a Structured Query Language based server...." (Fogel, 1992). Omnis 7 has the capability to read several different file formats facilitating the importation of data from other programs, foreign files or databases (Blyth Software, 1990).

Omnis 7's database engine allows dozens of files to be open simultaneously, can display fields from multiple files on a single window and can update all open files with a single command. In addition, Omnis 7 offers

> "...one of the most powerful GUI builders in the industry. To create a data entry window, the developer paints it in design mode, and Omnis 7 makes it come alive in execute mode. Aside from data-entry fields, virtually any standard GUI element, such as push buttons, check boxes, and scrolling lists, can be incorporated into an Omnis 7 window. What's more, multiple windows can be open simultaneously, each one movable, scrollable and sizeable, and each operating independently or in sync with the others." (Fogel, 1992)

For the potential programmer/implementor, Blyth Software Inc.'s Omnis 7 presents speed and virtually unlimited flexibility in the possible implementation of the Naval Postgraduate School Scheduling System. Furthermore, Omnis 7 can be found in the NPS software inventory which may preclude its purchase for any follow-on work to NPS^3 (Ragan, 1992).

APPENDIX A: CURRENT NPS^3 PHYSICAL DATA FLOW DIAGRAMS

In the following figures, FIG. A-1 thru FIG. A-6, the physical context data flow diagrams and the physical systems level data flow diagrams for the current NPS Scheduling System are presented so that the reader may obtain a more detailed understanding of what the current system does and how the current system is implemented.

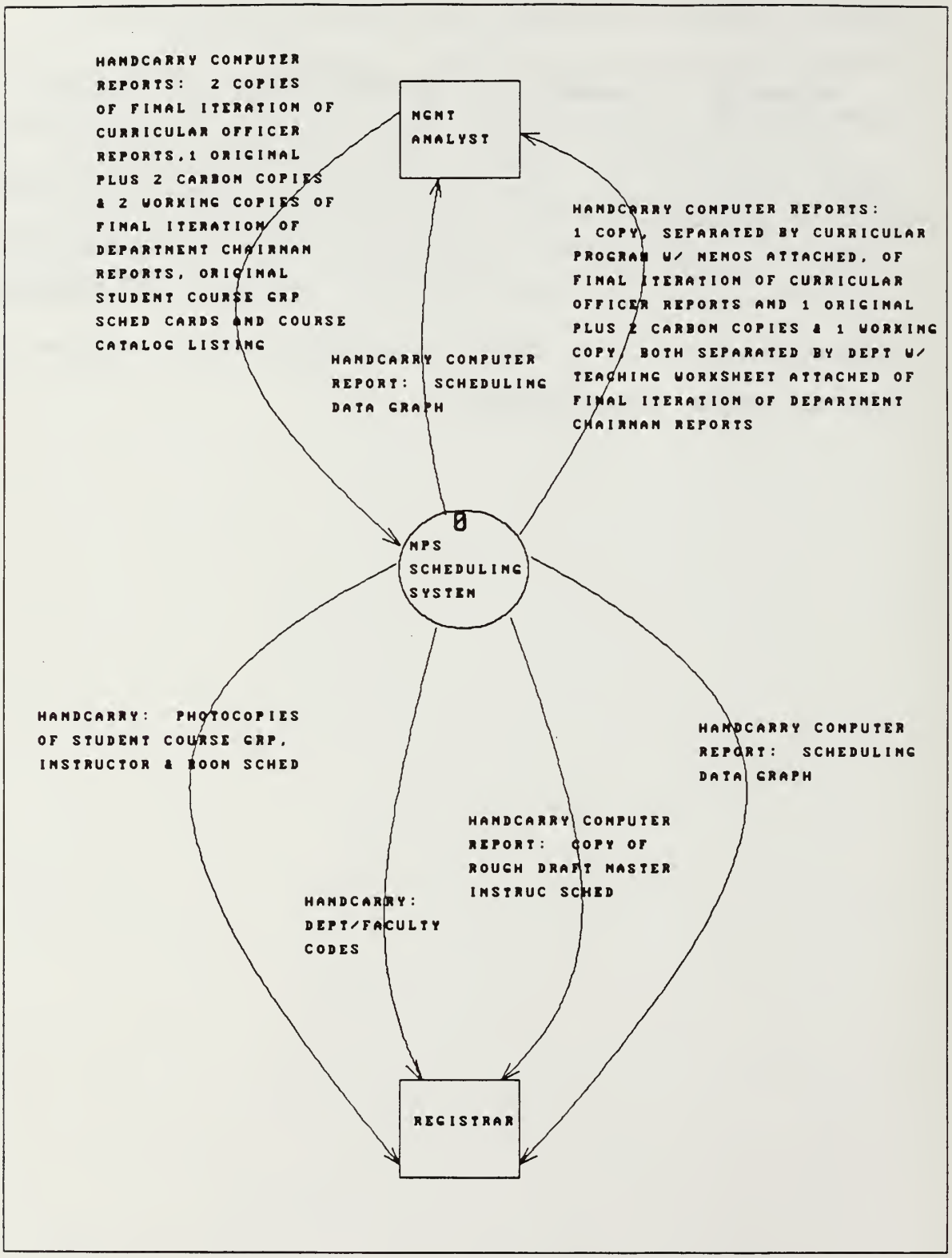

HANDCARRY COMPUTER REPORTS: 2 COPIES OF FINAL ITERATION OF CURRICULAR OFFICER REPORTS,1 ORIGINAL PLUS 2 CARBON COPIES & 2 WORKING COPIES OF FINAL ITERATION OF DEPARTMENT CHAIRMAN REPORTS, ORIGINAL STUDENT COURSE GRP SCHED CARDS AND COURSE CATALOG LISTING
MGMT ANALYST
HANDCARRY COMPUTER REPORTS: 1 COPY, SEPARATED BY CURRICULAR PROGRAM W/ MEMOS ATTACHED, OF FINAL ITERATION OF CURRICULAR OFFICER REPORTS AND 1 ORIGINAL PLUS 2 CARBON COPIES & 1 WORKING COPY, BOTH SEPARATED BY DEPT W/ TEACHING WORKSHEET ATTACHED OF FINAL ITERATION OF DEPARTMENT CHAIRMAN REPORTS
HANDCARRY COMPUTER REPORT: SCHEDULING DATA GRAPH
0
NPS SCHEDULING SYSTEM
HANDCARRY: PHOTOCOPIES OF STUDENT COURSE GRP, INSTRUCTOR & ROOM SCHED
HANDCARRY COMPUTER REPORT: SCHEDULING DATA GRAPH
HANDCARRY COMPUTER REPORT: COPY OF ROUGH DRAFT MASTER INSTRUC SCHED
HANDCARRY: DEPT/FACULTY CODES
REGISTRAR

FIG. A-1, Physical Context Dataflow Diagram of the current NPS Scheduling System.

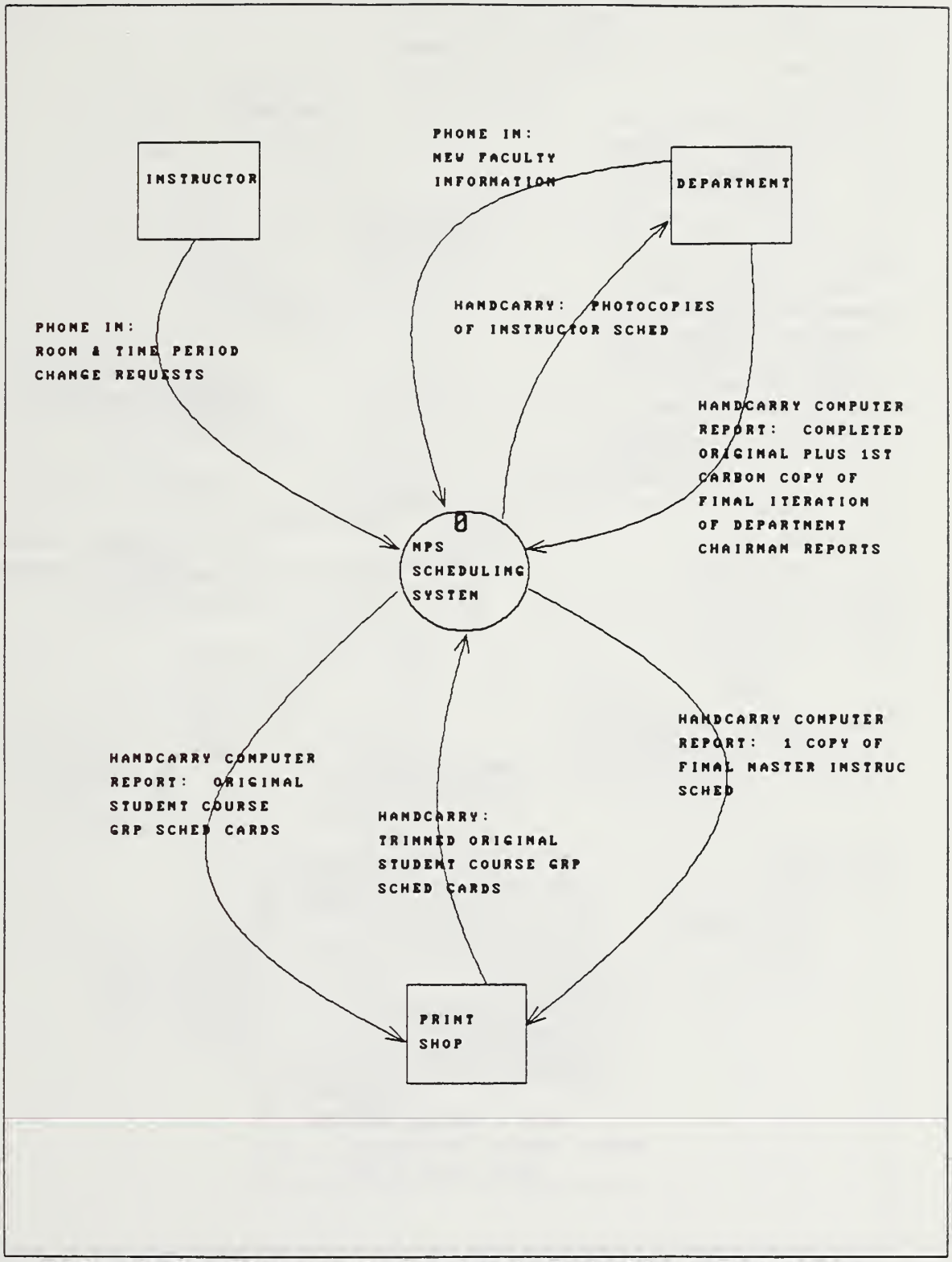


FIG. A-2, Physical Context Dataflow Diagram of the current NPS Scheduling System, cont'd.

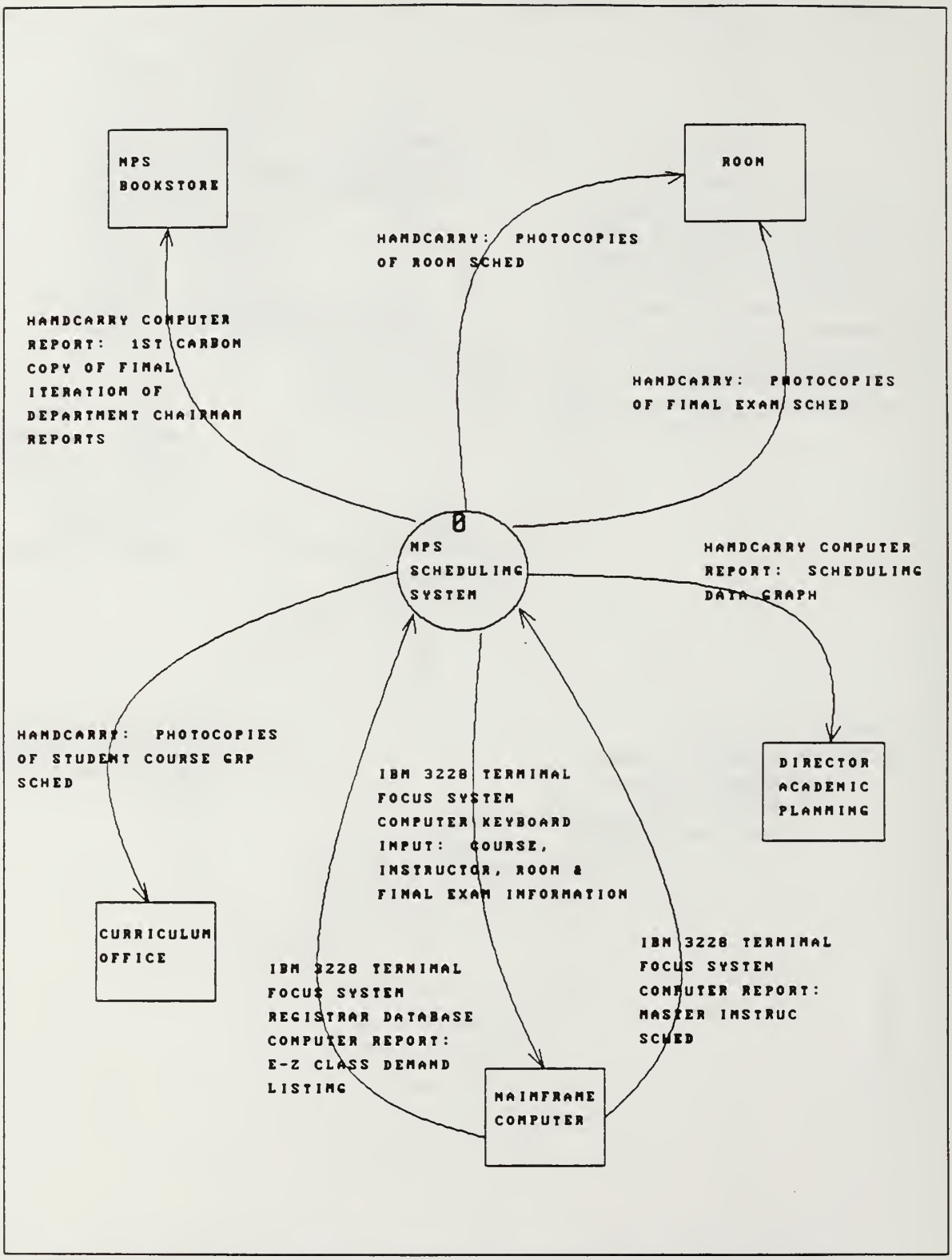


FIG. A-3, Physical Context Dataflow Diagram of the current NPS Scheduling System, cont'd.

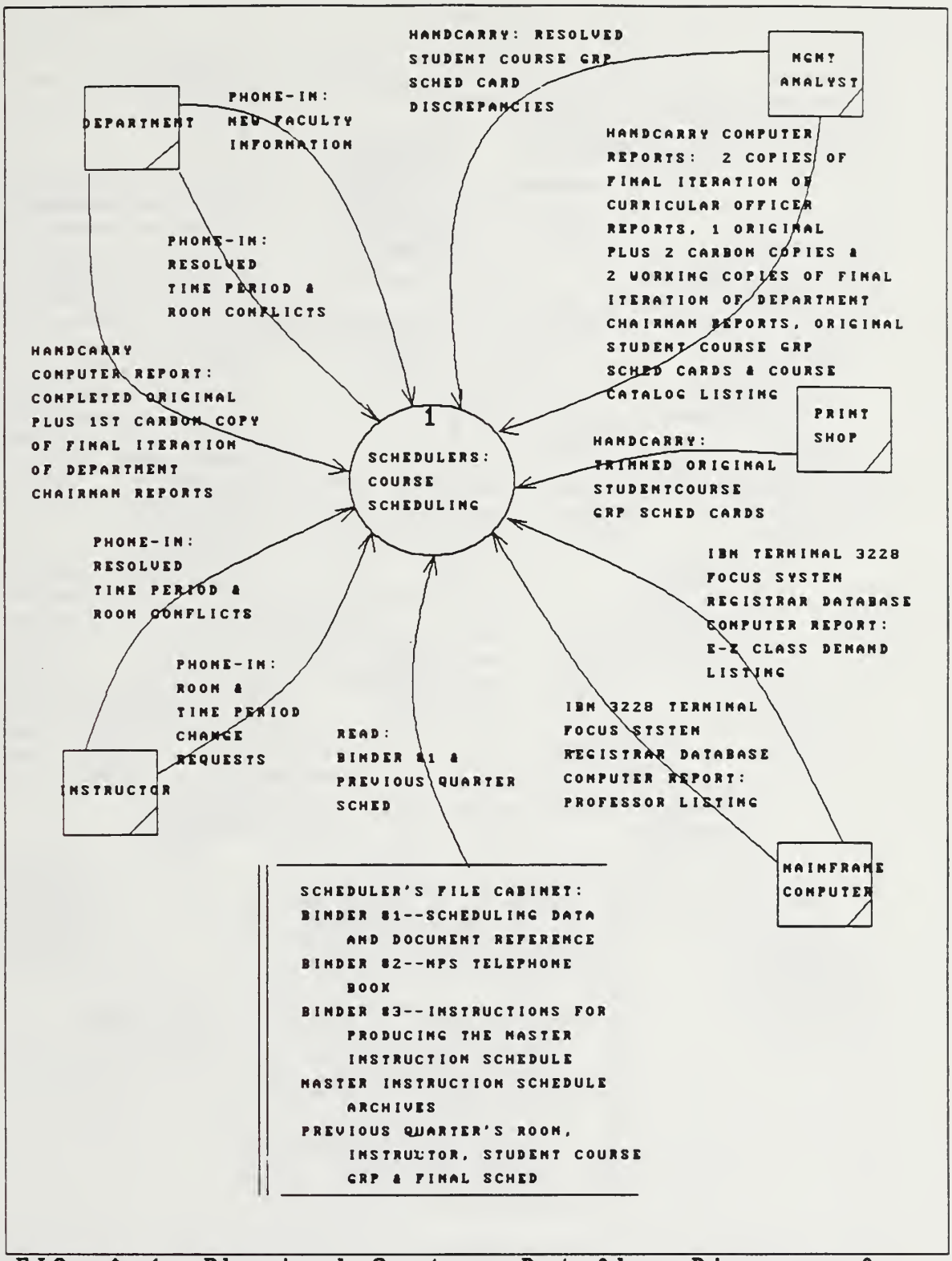


FIG. A-4, Physical Systems Dataflow Diagram of inputs to the Course Scheduling Process.

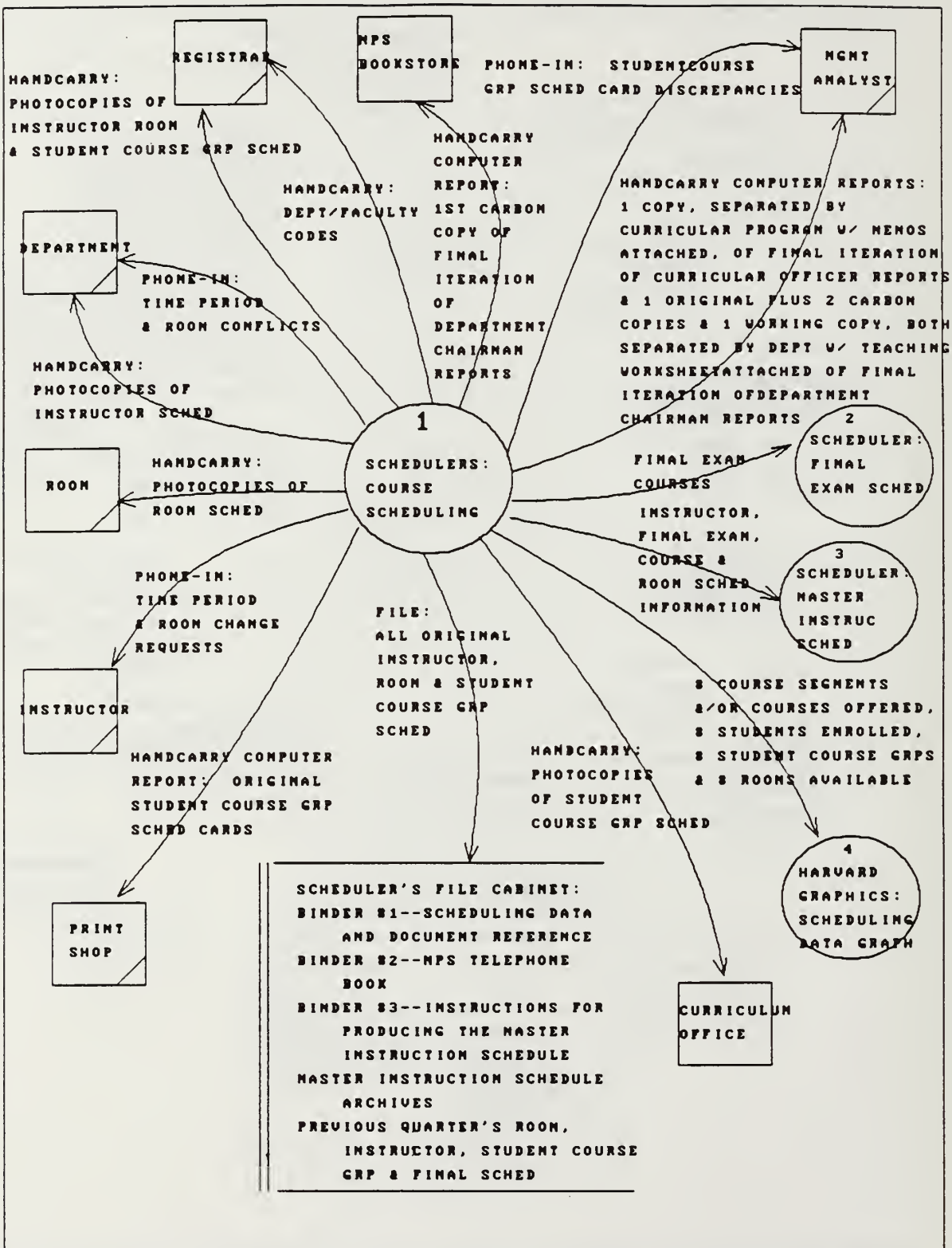

REGISTRAR
NPS BOOKSTORE
MGMT ANALYST
DEPARTMENT
ROOM
INSTRUCTOR
PRINT SHOP
CURRICULUM OFFICE
1
SCHEDULERS: COURSE SCHEDULING
2
SCHEDULER: FINAL EXAM SCHED
3
SCHEDULER: MASTER INSTRUC SCHED
4
HARVARD GRAPHICS: SCHEDULING DATA GRAPH
HANDCARRY: PHOTOCOPIES OF INSTRUCTOR ROOM & STUDENT COURSE GRP SCHED
HANDCARRY: DEPT/FACULTY CODES
HANDCARRY COMPUTER REPORT: 1ST CARBON COPY OF FINAL ITERATION OF DEPARTMENT CHAIRMAN REPORTS
PHONE-IN: STUDENT COURSE GRP SCHED CARD DISCREPANCIES
HANDCARRY COMPUTER REPORTS: 1 COPY, SEPARATED BY CURRICULAR PROGRAM W/ MEMOS ATTACHED, OF FINAL ITERATION OF CURRICULAR OFFICER REPORTS & 1 ORIGINAL PLUS 2 CARBON COPIES & 1 WORKING COPY, BOTH SEPARATED BY DEPT W/ TEACHING WORKSHEET ATTACHED OF FINAL ITERATION OF DEPARTMENT CHAIRMAN REPORTS
PHONE-IN: TIME PERIOD & ROOM CONFLICTS
HANDCARRY: PHOTOCOPIES OF INSTRUCTOR SCHED
HANDCARRY: PHOTOCOPIES OF ROOM SCHED
FINAL EXAM COURSES INSTRUCTOR, FINAL EXAM, COURSE & ROOM SCHED INFORMATION
PHONE-IN: TIME PERIOD & ROOM CHANGE REQUESTS
FILE: ALL ORIGINAL INSTRUCTOR, ROOM & STUDENT COURSE GRP SCHED
COURSE SEGMENTS #/OR COURSES OFFERED, # STUDENTS ENROLLED, # STUDENT COURSE GRPS & # ROOMS AVAILABLE
HANDCARRY COMPUTER REPORT: ORIGINAL STUDENT COURSE GRP SCHED CARDS
HANDCARRY: PHOTOCOPIES OF STUDENT COURSE GRP SCHED
SCHEDULER'S FILE CABINET:
BINDER #1--SCHEDULING DATA AND DOCUMENT REFERENCE
BINDER #2--NPS TELEPHONE BOOK
BINDER #3--INSTRUCTIONS FOR PRODUCING THE MASTER INSTRUCTION SCHEDULE
MASTER INSTRUCTION SCHEDULE ARCHIVES
PREVIOUS QUARTER'S ROOM, INSTRUCTOR, STUDENT COURSE GRP & FINAL SCHED

FIG. A-5, Physical Systems Dataflow Diagram of outputs from the Course Scheduling Process.

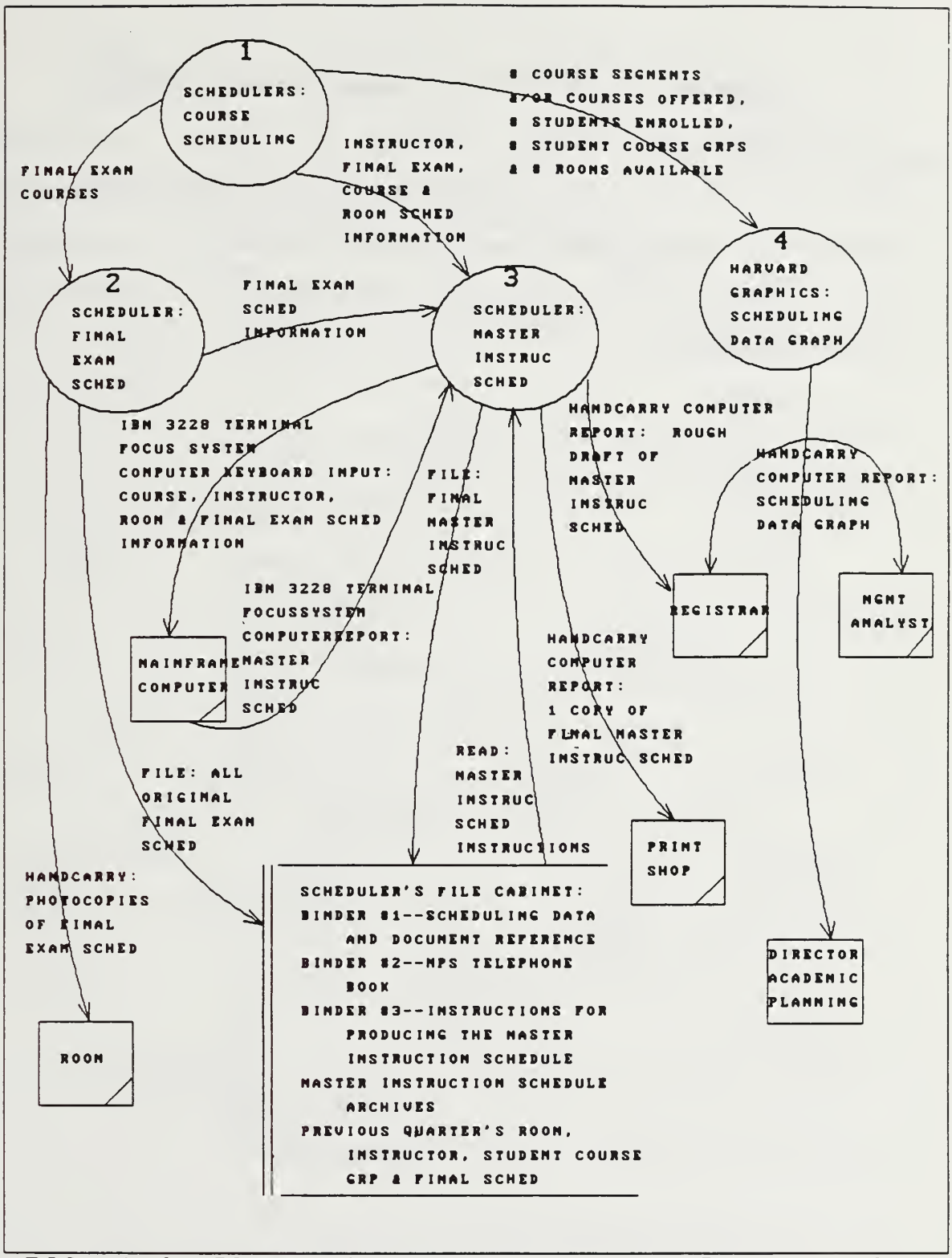

1
SCHEDULERS: COURSE SCHEDULING
COURSE SEGMENTS
&/OR COURSES OFFERED,
STUDENTS ENROLLED,
STUDENT COURSE GRPS
& # ROOMS AVAILABLE
FINAL EXAM COURSES
INSTRUCTOR, FINAL EXAM, COURSE & ROOM SCHED INFORMATION
2
SCHEDULER: FINAL EXAM SCHED
FINAL EXAM SCHED INFORMATION
3
SCHEDULER: MASTER INSTRUC SCHED
4
HARVARD GRAPHICS: SCHEDULING DATA GRAPH
IBM 3228 TERMINAL FOCUS SYSTEM COMPUTER KEYBOARD INPUT: COURSE, INSTRUCTOR, ROOM & FINAL EXAM SCHED INFORMATION
FILE: FINAL MASTER INSTRUC SCHED
HANDCARRY COMPUTER REPORT: ROUGH DRAFT OF MASTER INSTRUC SCHED
HANDCARRY COMPUTER REPORT: SCHEDULING DATA GRAPH
IBM 3228 TERMINAL FOCUSSYSTEM COMPUTEREEPORT: MASTER INSTRUC SCHED
MAINFRAME COMPUTER
REGISTRAR
MGMT ANALYST
HANDCARRY COMPUTER REPORT: 1 COPY OF FINAL MASTER INSTRUC SCHED
READ: MASTER INSTRUC SCHED INSTRUCTIONS
FILE: ALL ORIGINAL FINAL EXAM SCHED
PRINT SHOP
HANDCARRY: PHOTOCOPIES OF FINAL EXAM SCHED
SCHEDULER'S FILE CABINET:
BINDER #1--SCHEDULING DATA AND DOCUMENT REFERENCE
BINDER #2--NPS TELEPHONE BOOK
BINDER #3--INSTRUCTIONS FOR PRODUCING THE MASTER INSTRUCTION SCHEDULE
MASTER INSTRUCTION SCHEDULE ARCHIVES
PREVIOUS QUARTER'S ROOM, INSTRUCTOR, STUDENT COURSE GRP & FINAL SCHED
DIRECTOR ACADEMIC PLANNING
ROOM

FIG. A-6, Physical Systems Dataflow Diagram of the remaining NPS Scheduling System Processes.

APPENDIX B: NPS^3 ENTITY RELATIONSHIP DIAGRAMS

The following report presents the Entity Relationship Diagrams, FIG. B-1, for the proposed Naval Postgraduate School Scheduling System (NPS^3). Following FIG. B-1 are the Entity definitions followed with the definitions for each of the Entity Attributes.

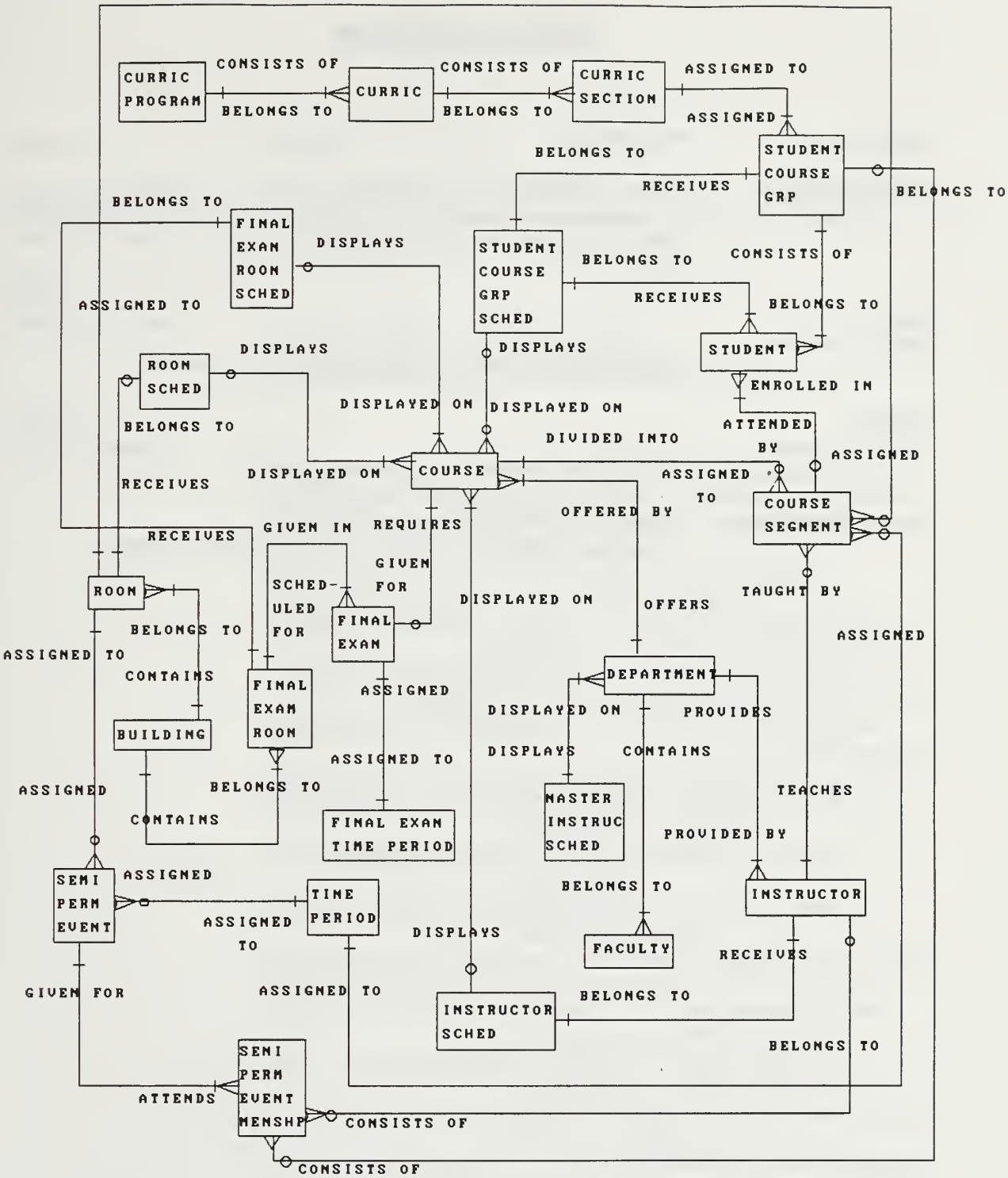


FIG. B-1, NPS Scheduling System Entity Relationship Diagram.

ENTITY DEFINITION

Entity: **BUILDING**

Description: This identifies the building of which classrooms, laboratories and final exam rooms are located in during an academic quarter at NPS.

Key Attributes:

BUILDING_CODE; Building_letter_code

Other Attributes:

BUILDING_NAME; Building_name

Relationships: Always CONTAINS many ROOMs.
Always CONTAINS many FINAL_EXAM_ROOMs.

ENTITY DEFINITION

Entity: **COURSE**

Description: This identifies the entire list of courses available at NPS based on the E-Z Demand List.

Key Attributes:

COURSE_TYPE; Course_type
COURSE_NUMBER; Course_number_code

Other Attributes:

FINAL_EXAM; Final_exam_course_indicator
REFRESHER COURSE; Refresher_course_indicator
NO_OF_SEGMENTS; Total_number_of_course_segments
LECT_HRS; Course_lecture_credit_hours
LAB_HRS; Course_lab_credit_hours
COURSE_REQUIRED_TIME; Scheduled_required_timeperiod_for_course_indicator
INSTRUC_REQUIRED_TIME; Scheduled_required timeperiod_for_instructor_indicator
DIVERSITY_OF_MAJORS; Diversity_of_majors_indicator
TECH_LAB; Technical_lab_indicator
SIMULTANEOUS_SCHED_COURSE; Simultaneously_scheduled_course
ACCELERATED_COURSE; Accelerated_course_indicator
SPECIAL_ROOM_REQUIREMENT; Special_room_requirements
REQUIRED_LAB_ROOM; Required_laboratory_room
DEPT_LTR_CODE^{FK}; Department_letter_code
FIN_EX_RM_CARD_NUMBER^{FK}; Final_exam_room schedule_card_number
SCG_CARD_NUMBER^{FK}; Student_course_group_schedule_card_number
INSTRUC_CARD_NUMBER^{FK}; Instructor_schedule_card_number
ROOM_CARD_NUMBER^{FK}; Room_schedule_card_number
QTR_NUMBER^{FK}; Schedule_quarter_number
YEAR^{FK}; Schedule_year
COMMENTS; Scheduler_course_comments
STATUS; Course_schedule_status

Relationships: Sometimes DIVIDED_INTO many COURSE_SEGMENTs.
Always OFFERED_BY one DEPARTMENT.
Sometimes DISPLAYED_ON one INSTRUCTOR_SCHED.
Sometimes REQUIRES one FINAL_EXAM.
Sometimes DISPLAYED_ON one ROOM_SCHED.
Sometimes DISPLAYED_ON one
FINAL_EXAM_ROOM_SCHED.
Sometimes DISPLAYED_ON one
STUDENT_COURSE_GRP_SCHED.

ENTITY DEFINITION

Entity: COURSE_SEGMENT

Description: This identifies all courses and their corresponding course segments.

Key Attributes:

COURSE_TYPE[FK]; Course_type
COURSE_NUMBER[FK]; Course_number_code
SEGMENT_NUMBER; Course_segment_number

Other Attributes:

SEGMENT_STUDENTS; Course_segment_number_of_students
LAB; Lab_indicator
TEACHING_TEAM; Teaching_team_indicator
DESIGNATED_STUDENT_COURSE_GRP; Designated_student_course_group_for_course_segment
BUILDING_CODE[FK]; Building_letter_code
ROOM_NUMBER[FK]; Room_number_in_building
DAY[FK]; Academic_day
PERIOD[FK]; Academic_hour
FACULTY_CODE[FK]; Department_faculty_code
DEPT_LTR_CODE[FK]; Department_letter_code
STATUS; Course_schedule_status

Relationships:

Always ASSIGNED_TO one COURSE.
Always ASSIGNED one ROOM.
Always ASSIGNED one TIME_PERIOD.
Always TAUGHT_BY one INSTRUCTOR.
Always ATTENDED_BY many STUDENTs.

ENTITY DEFINITION

Entity: **CURRIC**

Description: Uniquely identifies a specific area of study, at NPS, within a curriculum program.

Key Attributes: CURRIC_LTR_CODE; Curriculum_letter_code

Other Attributes:

PROGRAM_LTR_CODE^{FK}; Curriculum_program_letter_code

Relationships: Always CONSISTS_OF many CURRIC_SECTIONs.
Always BELONGS_TO one CURRIC_PROGRAM.

ENTITY DEFINITION

Entity: **CURRIC_PROGRAM**

Description: Uniquely identifies an overall area study, at NPS, that may encompass many curriculums.

Key Attributes: PROGRAM_LTR_CODE; Curriculum_program_letter_code

Other Attributes:

Relationships: Always CONSISTS_OF many CURRICulums.

ENTITY DEFINITION

Entity: **CURRIC_SECTION**

Description: Uniquely identifies a section of students assigned to the same curriculum.

Key Attributes:

CURRIC_LTR_CODE^{FK}; Curriculum_letter_code
SECTION_NUMBER; Curriculum_section_number

Other Attributes:

Relationships: Always BELONGS_TO one CURRICulum.
Always ASSIGNED_TO many STUDENT_COURSE_GRPs.

ENTITY DEFINITION

Entity: **DEPARTMENT**

Description: This identifies the research/teaching organizational unit at NPS.

Key Attributes: DEPT_LTR_CODE; Department_letter_code

Other Attributes:

DEPT_NAME; Department_name
MSTR_INSTR_SCHD_QTR_NAME[FK];
Master_instruction_schedule_quarter_name
MSTR_INSTR_SCHD_YEAR[FK];
Master_instruction_schedule_year

Relationships: Always OFFERS many COURSEs.
Always PROVIDES many INSTRUCTORs.
Always CONTAINS many FACULTY.
Always DISPLAYED_ON one MASTER_INSTRUC_SCHED.

ENTITY DEFINITION

Entity: **FACULTY**

Description: This identifies the members of a particular department.

Key Attributes:

DEPT_LTR_CODE[FK]; Department_letter_code
FACULTY_CODE; Department_faculty_code

Other Attributes:

FACULTY_NAME; Faculty_name
ACAD_COUNCIL_MBRSHP; Academic_council_membership_indicator
FAC_COUNCIL_OFF_LIST; Faculty_council_officers_listing_indicator
ASW_MBRSHP; ASW_academic_group_indicator
EW_MBRSHP: EW_academic_group_indicator
C3_MBRSHP; C3_academic_group_indicator
SSAG_MBRSHP; Space_systems_academic_group_indicator

Academic Groups?

Relationships: Always BELONGS_TO one DEPARTMENT.

ENTITY DEFINITION

Entity: **FINAL_EXAM**

Description: This identifies a course and/or course segment that requires a final exam.

Key Attributes:
COURSE_TYPE^{FK}; Course_type
COURSE_NUMBER^{FK}; Course_number_code
SEGMENT_NUMBER^{FK}; Course_segment_number

Other Attributes:
BUILDING_CODE^{FK}; Building_letter_code
FINAL_EXAM_ROOM_NUMBER^{FK}; Final_exam_room_number_in_building
FINAL_EXAM_DAY^{FK}; Final_exam_academic_day
FINAL_EXAM_PERIOD^{FK}; Final_exam_academic_hour

Relationships:
Always GIVEN_IN one FINAL_EXAM_ROOM.
Always GIVEN_FOR one COURSE.
Always ASSIGNED one FINAL_EXAM_TIME_PERIOD.

ENTITY DEFINITION

Entity: **FINAL_EXAM_ROOM**

Description: This identifies the building and room in which a final exam for a course and/or course segment is given in.

Key Attributes:

BUILDING_CODE^{FK}; Building_letter_code
FINAL_EXAM_ROOM_NUMBER; Final_exam_room_number_in_building

Other Attributes:

CAPACITY; Room_seating_capacity
ARRANGEMENT; Room_seating_arrangements
FEATURES; Room_features_or_equipment
TYPE; Room_type
FIN_EX_RM_CARD_NUMBER^{FK}; Final_exam_room_schedule_card_number

Relationships: Always SCHEDULED_FOR many FINAL_EXAMs.
Always BELONGS_TO one BUILDING.
Always RECEIVES one FINAL_EXAM_ROOM_SCHED.

ENTITY DEFINITION

Entity: **FINAL_EXAM_ROOM_SCHED**

Description: This identifies the final exam schedule for a final exam room.

Key Attributes:

FIN_EX_RM_CARD_NUMBER; Final_exam_room_schedule_card_number
QTR_NUMBER; Schedule_quarter_number
YEAR; Schedule_year

Other Attributes:

BUILDING_CODE^{FK}; Building_letter_code
FINAL_EXAM_ROOM_NUMBER^{FK}; Final_exam_room_number_in_building

Relationships: Always DISPLAYS many COURSEs.
Always BELONGS_TO one FINAL_EXAM_ROOM.

ENTITY DEFINITION

Entity: **FINAL_EXAM_TIME_PERIOD**

Description: This identifies the scheduled day and time that a final exam is given in a final exam room.

Key Attributes:
FINAL_EXAM_DAY; Final_exam_academic_day
FINAL_EXAM_PERIOD; Final_exam_academic_hour

Other Attributes:
COURSE_TYPE^{FK}; Course_type
COURSE_NUMBER^{FK}; Course_number_code
SEGMENT_NUMBER^{FK}; Course_segment_number

Relationships: Always ASSIGNED_TO one FINAL_EXAM.

ENTITY DEFINITION

Entity: **INSTRUCTOR**

Description: This identifies the person who teaches a particular course and/or course segment.

Key Attributes:

DEPT_LTR_CODE[FK]; Department_letter_code
FACULTY_CODE; Department_faculty_code

Other Attributes:

INSTRUC_NAME; Instructor_name
DEPT_CHAIR_OR_DEAN; Department_chairman_or_dean_indicator
ACAD_COUNCIL_MBRSHP; Academic_council_membership_indicator
FAC_COUNCIL_OFF_LIST; Faculty_council_officers_listing_indicator
ASW_MBRSHP; ASW_academic_group_indicator
EW_MBRSHP: EW_academic_group_indicator
C3_MBRSHP; C3_academic_group_indicator
SSAG_MBRSHP; Space_systems_academic_group_indicator
INSTRUC_CARD_NUMBER[FK]; Instructor_schedule_card_number

Relationships: Always PROVIDED_BY one DEPARTMENT.
Sometimes TEACHES many COURSE_SEGMENTs.
Sometimes BELONGS_TO many SEMI_PERM_EVENT_MEMSHPs.
Always RECEIVES one INSTRUCTOR_SCHED.

ENTITY DEFINITION

Entity: **INSTRUCTOR_SCHED**

Description: This identifies the schedule for an Instructor during a given quarter.

Key Attributes:

INSTRUC_CARD_NUMBER; Instructor_schedule_card_number
QTR_NUMBER; Schedule_quarter_number
YEAR; Schedule_year

Other Attributes:

DEPT_LTR_CODE^{FK}; Department_letter_code
FACULTY_CODE^{FK}; Department_faculty_code
INSTRUC_NAME; Instructor_name

Relationships: Always BELONGS_TO one INSTRUCTOR.
Always DISPLAYS many COURSEs.

ENTITY DEFINITION

Entity: **MASTER_INSTRUC_SCHED**

Description: This identifies the schedule for all courses offered during a given quarter.

Key Attributes:

MSTR_INSTR_SCHD_QTR_NAME;Master_instruction_schedule_quarter_name
MSTR_INSTR_SCHD_YEAR; Master_instruction_schedule_year

Other Attributes:

Relationships: Always DISPLAYS many DEPARTMENTs.

ENTITY DEFINITION

Entity: **ROOM**

Description: This identifies a classroom or a laboratory that a course or course segment is taught in.

Key Attributes:

BUILDING_CODE^{FK}; Building_letter_code
ROOM_NUMBER; Room_number_in_building

Other Attributes:

CAPACITY; Room_seating_capacity
ARRANGEMENT; Room_seating_arrangements
FEATURES; Room_features_or_equipment
TYPE; Room_type
ROOM_CARD_NUMBER; Room_schedule
_card_number

Relationships:

Sometimes ASSIGNED_TO many COURSE_SEGMENTs.
Sometimes RECEIVES one ROOM_SCHED.
Always BELONGS_TO one BUILDING.
Sometimes ASSIGNED_TO many SEMI_PERM_EVENTs.

ENTITY DEFINITION

Entity: ROOM_SCHED

Description: This identifies the schedule for a given classroom or laboratory.

Key Attributes:

ROOM_CARD_NUMBER; Room_schedule_card_number
QTR_NUMBER; Schedule_quarter_number
YEAR; Schedule_year

Other Attributes:

BUILDING_CODE^{FK}; Building_letter_code
ROOM_NUMBER^{FK}; Room_number_in_building

Relationships: Always BELONGS_TO one ROOM.
Always DISPLAYS many COURSEs.

ENTITY DEFINITION

Entity: **SEMI_PERM_EVENT**

Description: This identifies an event that is scheduled during the same time period every quarter.

Key Attributes:

EVENT_TYPE; Semi_permanent_event_type
EVENT_NAME; Semi_permanent_event_name

Other Attributes:

BUILDING_CODE^{FK}; Building_letter_code
ROOM_NUMBER^{FK}; Room_number_in_building
DAY^{FK}; Academic_day
PERIOD^{FK}; Academic_hour

Relationships: Always ASSIGNED one ROOM.
Always GIVEN_FOR many SEMI_PERM_EVENT_MEMSHPs.
Always ASSIGNED one TIME_PERIOD.

ENTITY DEFINITION

Entity: **SEMI_PERM_EVENT_MEMSHP**

Description: This identifies the membership for a semi permanent event at NPS.

Key Attributes:

EVENT_TYPE[FK]; Semi_permanent_event_type
EVENT_NAME[FK]; Semi_permanent_event_name
FACULTY_CODE[FK]; Department_faculty_code
DEPT_LTR_CODE[FK]; Department_letter_code
CURRIC_LTR_CODE[FK]; Curriculum_letter_code
SECTION_NUMBER[FK]; Curriculum_section_number
GROUP_NUMBER[FK]; Curriculum_student
_course_group_number

Other Attributes:

Relationships: Always ATTENDS one SEMI_PERM_EVENT.
Sometimes CONSISTS_OF one INSTRUCTOR.
Sometimes CONSISTS_OF one STUDENT_COURSE_GRP.

ENTITY DEFINITION

Entity: **STUDENT**

Description: This identifies a person attending a course of instruction at NPS.

Key Attributes:

STUD_SSN; Student_social_security_number

Other Attributes:

STUD_NAME; Student_name
CURRIC_LTR_CODE^{FK}; Curriculum_letter_code
SECTION_NUMBER^{FK}; Curriculum_section_number
GROUP_NUMBER^{FK}; Curriculum_student
_course_group_number
SCG_CARD_NUMBER^{FK}; Student_course
_group_schedule_card_number
QTR_NUMBER^{FK}; Schedule_quarter_number
YEAR^{FK}; Schedule_year
COURSE_TYPE^{FK}; Course_type
COURSE_NUMBER^{FK}; Course_number_code
SEGMENT_NUMBER^{FK}; Course_segment_number

Relationships: Always BELONGS_TO one STUDENT_COURSE_GRP.
Always RECEIVES one STUDENT_COURSE_GRP_SCHED.
Sometimes ENROLLED_IN one COURSE_SEGMENT.

ENTITY DEFINITION

Entity: **STUDENT_COURSE_GRP**

Description: This identifies the group of students taking the same courses during a quarter at NPS.

Key Attributes:

CURRIC_LTR_CODE^{FK}; Curriculum_letter_code
SECTION_NUMBER^{FK}; Curriculum_section_number
GROUP_NUMBER; Curriculum_student_course_group_number

Other Attributes:

GROUP_STUDENTS; Curriculum_student_course_group_number_of_students
UNUSUAL_COURSE_COMBINATION; Unusual_course_combination_indicator

Relationships: Always ASSIGNED one CURRIC_SECTION.
Always CONSISTS_OF many STUDENTs (may consist of one student).
Sometimes BELONGS_TO many SEMI_PERM_EVENT_MEMSHPs.
Always RECEIVES one STUDENT_COURSE_GRP_SCHED.

ENTITY DEFINITION

Entity: **STUDENT_COURSE_GRP_SCHED**

Description: This identifies the schedule for students taking the same courses during a quarter at NPS.

Key Attributes:

SCG_CARD_NUMBER; Student_course_group_schedule_card_number
QTR_NUMBER; Schedule_quarter_number
YEAR; Schedule_year

Other Attributes:

CURRIC_LTR_CODE^{FK}; Curriculum_letter_code
SECTION_NUMBER^{FK}; Curriculum_section_number
GROUP_NUMBER^{FK}; Curriculum_student_course_group_number

Relationships: Always BELONGS_TO many STUDENTs (may belong to one student).
Sometimes DISPLAYS many COURSEs.
Always BELONGS_TO one STUDENT_COURSE_GRP.

ENTITY DEFINITION

Entity: **TIME_PERIOD**

Description: This identifies the day and time that a course, course segment or semi permanent event is scheduled.

Key Attributes:

DAY; Academic_day
PERIOD; Academic_hour

Other Attributes:

Relationships: Sometimes ASSIGNED_TO many SEMI_PERM_EVENTs.
Sometimes ASSIGNED_TO many COURSE_SEGMENTs.

ATTRIBUTE DEFINITIONS

Academic_council_membership_indicator:
    Text 1, Mask X
            where X is Y or N
    Indicates membership in Academic Council

Academic_day:
    Text 9
    Name of the day of the academic week

Academic_hour:
    Numeric 1
    Hour-long period during the academic day

Accelerated_course_indicator:
    Text 1, Mask X,
            where X is Y or N
    Indicates courses normally scheduled for double the
        course load

ASW_academic_group_indicator:
    Text 1, Mask X
            where X is Y or N
    Indicates membership in ASW academic group

Building_letter_code:
    Text 1
    First letter of the name of a particular academic
       building

Building_name:
    Text 20
    Name of an academic building

C3_academic_group_indicator:
    Text 1, Mask X
            where X is Y or N
    Indicates membership in C3 academic group

Course_lab_credit_hours:
    Numeric 1
    Number of credit hours assigned to the lab component
        of a course or its course segment

Course_lecture_credit_hours:
    Numeric 1
    Number of credit hours assigned to the lecture component
        of a course or its course segment

Course_number_code:
    Numeric 4, Mask NXXX
        where N is 0-4, XXX is course code
    Number code assigned to a course

Course_schedule_status_code:
    Text 4
    Status of the scheduling of a course

Course_segment_number:
    Numeric 2
    Number of the segment into which a course has been
        divided

Course_segment_number_of_students:
    Numeric 3
    Number of students assigned to a course segment

Course_type:
    Text 2
    Code for the type subject a course teaches

Curriculum_letter_code:
    Text 2
    Letter code for a curriculum within a curriculum program

Curriculum_program_letter_code:
    Text 2
    Letter code for a curriculum program

Curriculum_section_number:
    Numeric 2
    Number code for a section within a curriculum

Curriculum_student_course_group_number:
    Numeric 2
    Number of a student course group within a curriculum
        section

Curriculum_student_course_group_number_of_students:
    Numeric 2
    Number of students in a student course group

Department_chairman_or_dean_indicator:
    Text 1, Mask X
        where X is Y of N
    Indicates Instructor who ius Department chairman of Dean

Department_faculty_code:
    Text 2
    Code used to identify each faculty member

Department_letter_code:
Text 2
Alpha code used to identify each department

Department_name:
Text 25
Name of department

Designated_student_course_group_for_course_segment:
Text 6, Mask XX YY ZZ
where XX is curriculum letter code, YY is curriculum section number and ZZ is curriculum student course group number
Designates students to be placed in specific course segments

Diversity_of_majors_indicator:
Text 1, Mask X
where X is Y or N
Indicates that the course has a diversity of majors in its enrollment

EW_academic_group_indicator:
Text 1, Mask X
where X is Y or N
Indicates membership in EW academic group

Faculty_council_officers_ listing_indicator:
Text 1, Mask X
where X is Y or N
Indicates membership in Provost and Superintendent council

Faculty_name:
Text 15
First 13 letters of faculty member last name, 1st letter of faculty member first name and 1st letter of faculty member middle name
Name of faculty member

Final_exam_academic_day:
Text 9
Name of the day of final's week

Final_exam_academic_hour:
Text 1
Hour-long period during final's week

Final_exam_course_indicator:
Text 1, Mask X
where X is Y or N
Indicates final exam requirement for course

Final_exam_room_number_in_building:
Text 4, Mask NNNA
where NNA is the sequential number of a room and A is a letter designator for a subdivision of a room
Number assigned to a room in a building used for a final exam

Final_exam_room_schedule_card_number:
Numeric 4
Sequential number of a final exam schedule card in a set of final exam schedule cards

Instructor_name:
Text 15
First 13 letters of instructor last name, 1st letter of instructor first name and 1st letter of instructor middle name
Name of an instuctor

Instructor_schedule_card_number:
Numeric 4
Sequential number of an instructor schedule card in a set of instructor schedule cards

Lab_indicator:
Text 1, Mask X
where X is Y or N
Indicates that the particular course segment is a laboratory

Master_instruction_schedule_year:
Numeric 4
Year for which the Master Instruction schedule is written

Master_instruction_schedule_quarter_name:
Text 6
Name of the season at the beginning of the quarter of the academic year

Refresher_course_indicator:
Text 1, Mask X
where X is Y or N
Indicates a course is a refresher course

Required_laboratory_room:
Text 6, Mask X-NNNA
where X is first letter of building, NNN is sequential number of a room and A is letter designator for a room subdivision
Number assigned to a required laboratory room for a course lab

Room_features_or_equipment
    Text 20
    Description of features of qpuipment about a room that distinguish it from ther rooms

Room_number_in_building:
    Text 4, Mask NNNA
        where NNN is sequential number of a room, A is letter designator for a room subdivision

Room_schedule_card_number:
    Numeric 4
    Sequential number of a room schedule card in a set of room schedule cards

Room_seating_arrangements:
    Text 1, Mask X
        where X is values T(tables and chairs) or D(desks) seating arrangements for students taught in a room

Room_seating_capacity:
    Numeric 3
    Number of chairs with table space or desks or standing room; functional student capacity

Room_type:
    Text 2, Mask XX
        where XX is values CR (classroom) or LB (laboratory) or SC (secure class) or SL (secure lab)
    Specifies the type of room

Schedule_quarter_number:
    Numeric 1
    Sequential number of a quarter in a schedule year

Schedule_year:
    Numeric 4
    Year for which a schedule is written

Scheduled_required_timeperiod_for_instructor_indicator:
    Text 1, Mask X
        where X is Y or N
    Indicates that course has a required timeperiod for the instructor

Scheduled_required_timeperiod_for_course_indicator:
    Text 1, Mask X
        where X is Y or N
    Indicates that course has a required timeperiod

Scheduler_course_comments:
Text 50
Comments from course schedulers regarding course

Semi_permanent_event_name:
Text 10
First 10 letters of a semi permanent event name (e.g. Dept mtg)

Semi_permanent_event_type:
Text 2
Code for the type semi permanent event schedule

Simultaneously_scheduled_course:
Text 7, Mask XX-NYYY
where XX is course type, N is 0-4 and YYY is course code
Identifies a course that simultaneously scheduled with the course in question

Space_systems_academic_indicator:
Text 1, Mask X
where X is Y or N
Indicates membership in Space systems academic gruop

Special_room_requirements:
Text 20
Special room features that a given course requires (eg computor terminals, classified rooms, NSA map rooms)

Student_course_group_schedule_card_number:
Numeric 4
Sequential number of a student course group schedule card in a set of student course group schedule cards

Student_name:
Text 15
First 13 letters of student last name, first letter of student first name, first letter of student middle name

Student_social_security_number:
Numeric 9
Social security number of a person

Teaching_team_indicator:
Text 1, Mask X
where X is Y or N
Indicates a course is taught by a team of instructors

Technical_lab_indicator:
Text 1, Mask X
where X is Y or N
Indicates that a course labortary is technical in nature

Total_number_of_course_segments:
Numeric 1
Total number of course segments a course is divided into

Unusual_course_combination_indicator:
Text 1, Mask X
where X is Y or N
Indicates unusual course combinations for a student course group

APPENDIX C: NPS^3 LOGICAL DATA FLOW DIAGRAMS

In an attempt to gain a more detailed description of the NPS Scheduling System user requirements presented by Nolan and Youngblood, and to enhance possible alternative implementation solutions, the logical dataflow diagrams for NPS^3 are presented in FIG. C-1 thru FIG. C-38. The Demarco-Yourdon system for dataflow diagrams is used.

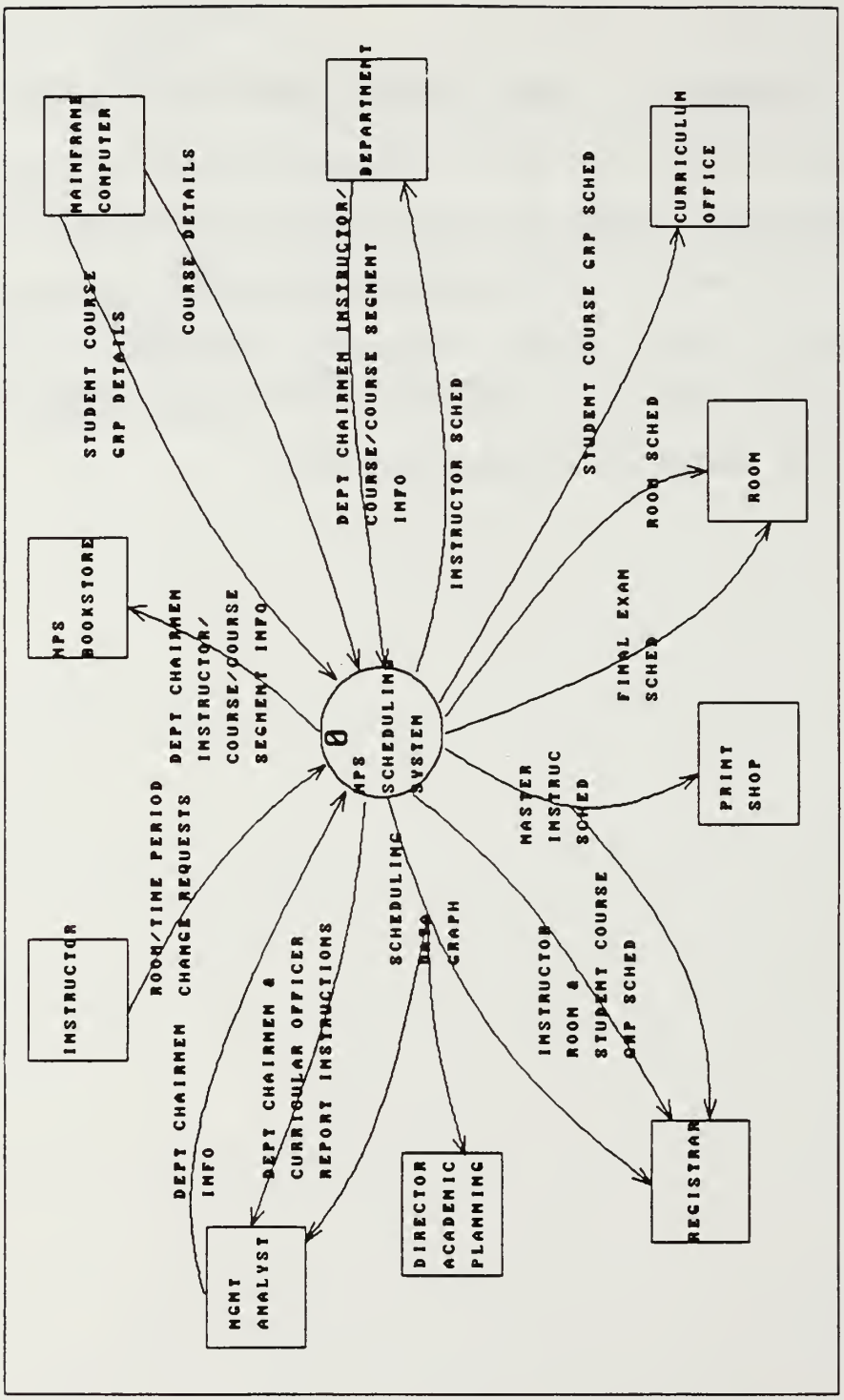


FIG. C-1, Logical Context Dataflow Diagram of the target NPS Scheduling System.

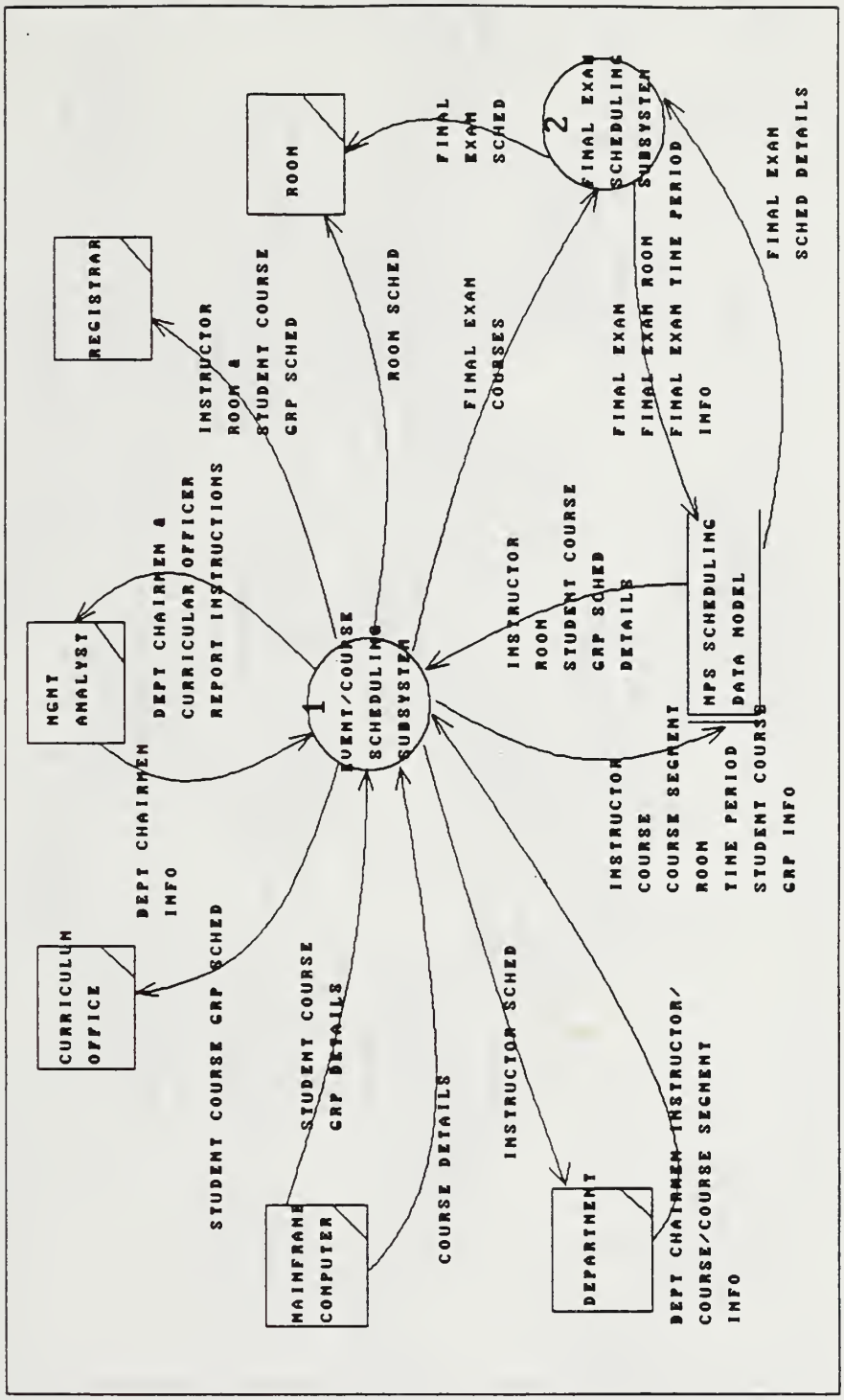


FIG. C-2, Logical Systems Dataflow Diagram of the target NPS Scheduling System.

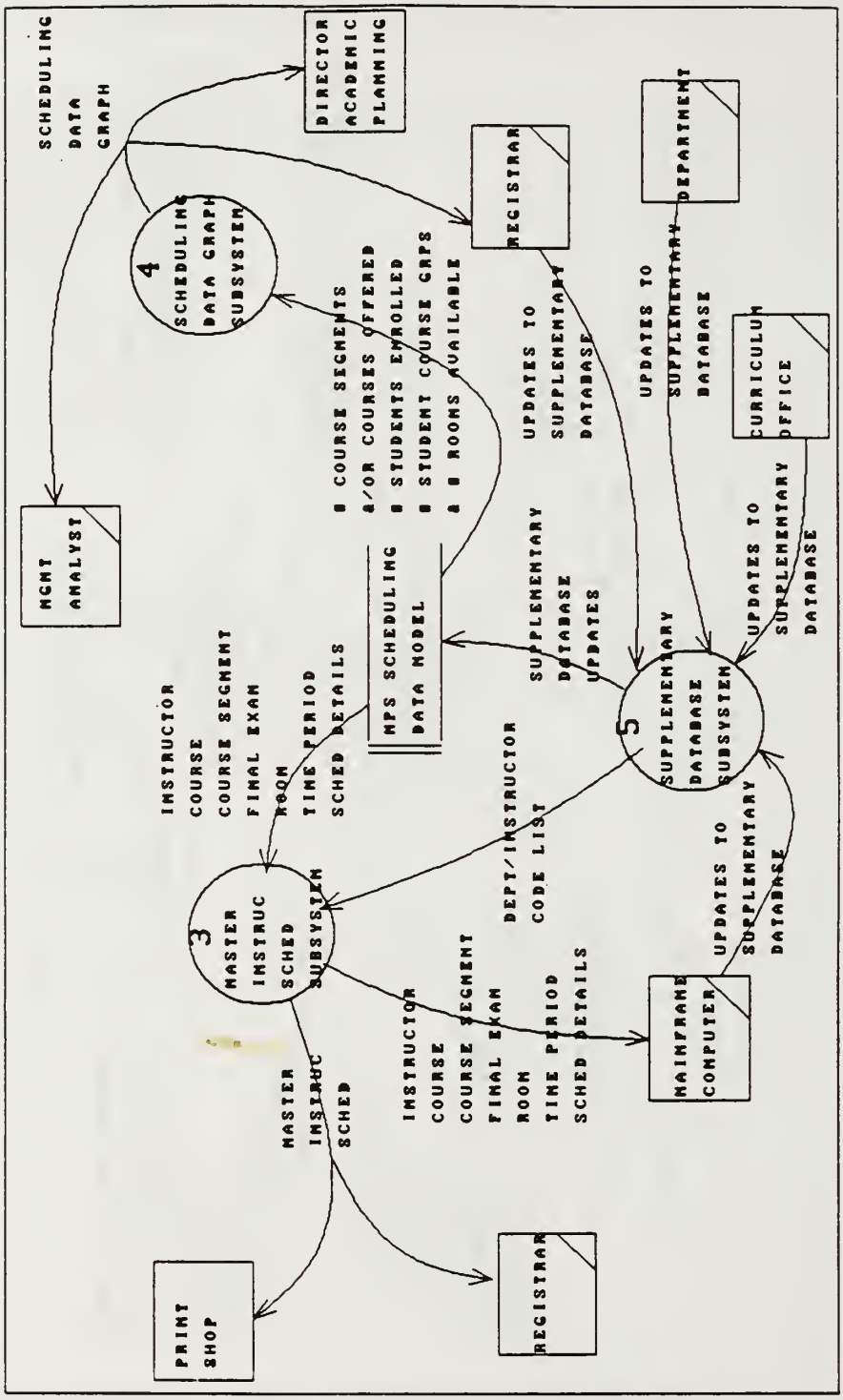


FIG. C-3, Logical Systems Dataflow Diagram of the target NPS Scheduling System, cont'd.

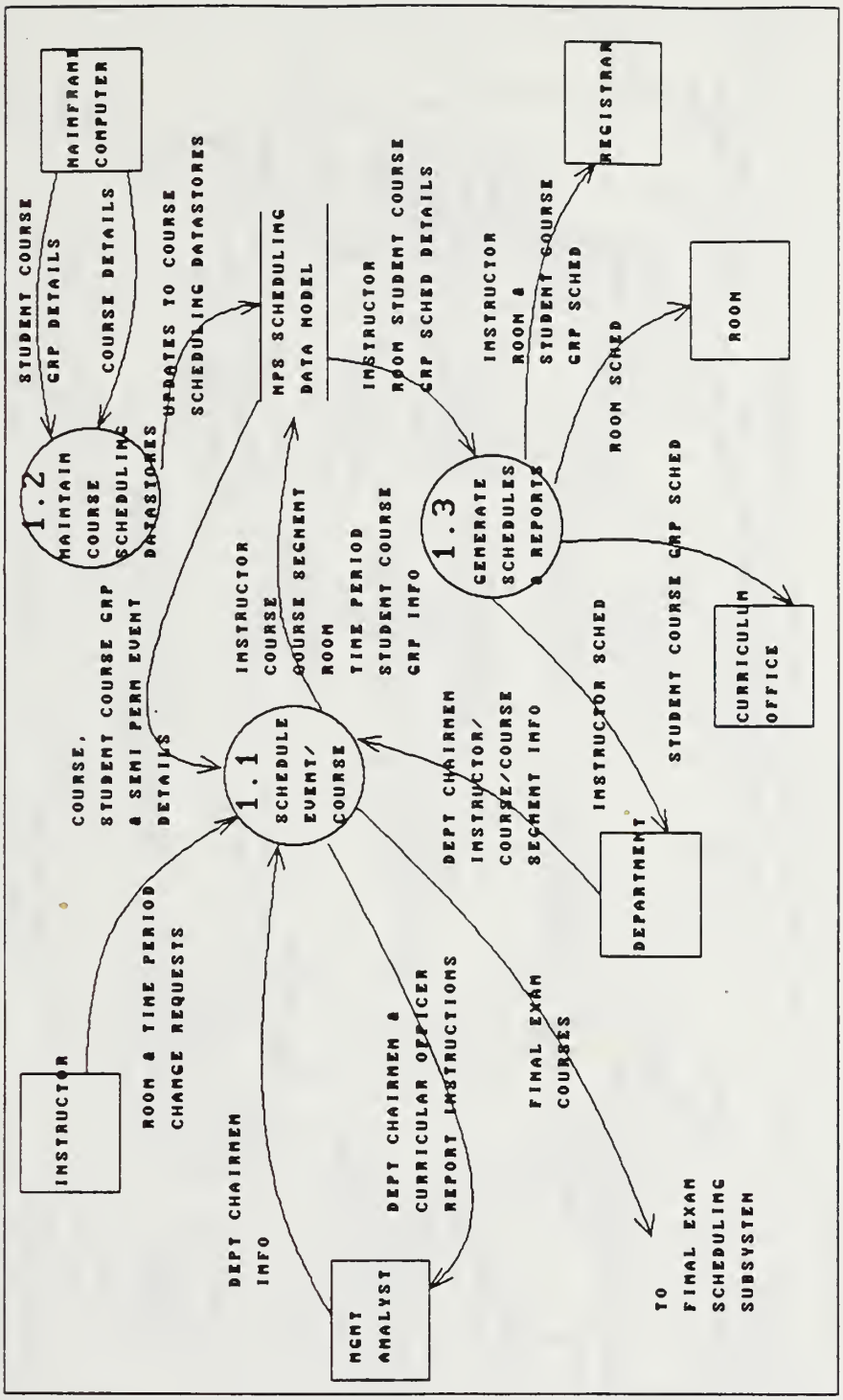

MAINFRAME COMPUTER
REGISTRAR
ROOM
CURRICULUM OFFICE
DEPARTMENT
INSTRUCTOR
MGMT ANALYST
1.1 SCHEDULE EVENT/ COURSE
1.2 MAINTAIN COURSE SCHEDULING DATASTORES
1.3 GENERATE SCHEDULES & REPORTS
STUDENT COURSE GRP DETAILS
COURSE DETAILS
UPDATES TO COURSE SCHEDULING DATASTORES
MPS SCHEDULING DATA MODEL
INSTRUCTOR ROOM STUDENT COURSE GRP SCHED DETAILS
INSTRUCTOR ROOM & STUDENT COURSE GRP SCHED
ROOM SCHED
STUDENT COURSE GRP SCHED
INSTRUCTOR SCHED
COURSE, STUDENT COURSE GRP & SEMI PERM EVENT DETAILS
INSTRUCTOR COURSE COURSE SEGMENT ROOM TIME PERIOD STUDENT COURSE GRP INFO
DEPT CHAIRMEN INSTRUCTOR/ COURSE/COURSE SEGMENT INFO
ROOM & TIME PERIOD CHANGE REQUESTS
DEPT CHAIRMEN INFO
DEPT CHAIRMEN & CURRICULAR OFFICER REPORT INSTRUCTIONS
FINAL EXAM COURSES
TO FINAL EXAM SCHEDULING SUBSYSTEM

FIG. C-4, Logical Dataflow Diagram for the target Event/Course Scheduling Subsystem.

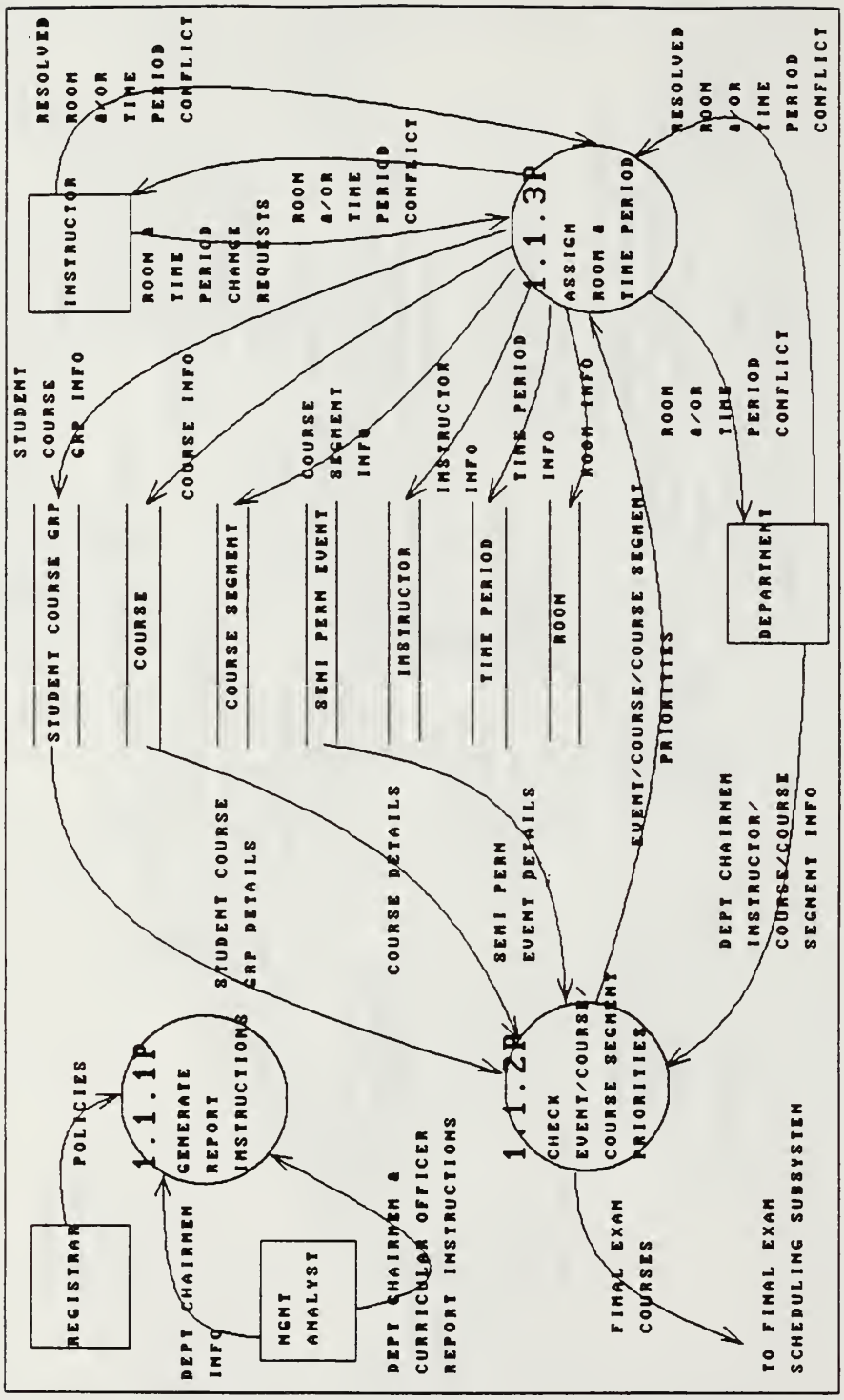


FIG. C-5, Logical Dataflow Diagram for Event/Course Scheduling.

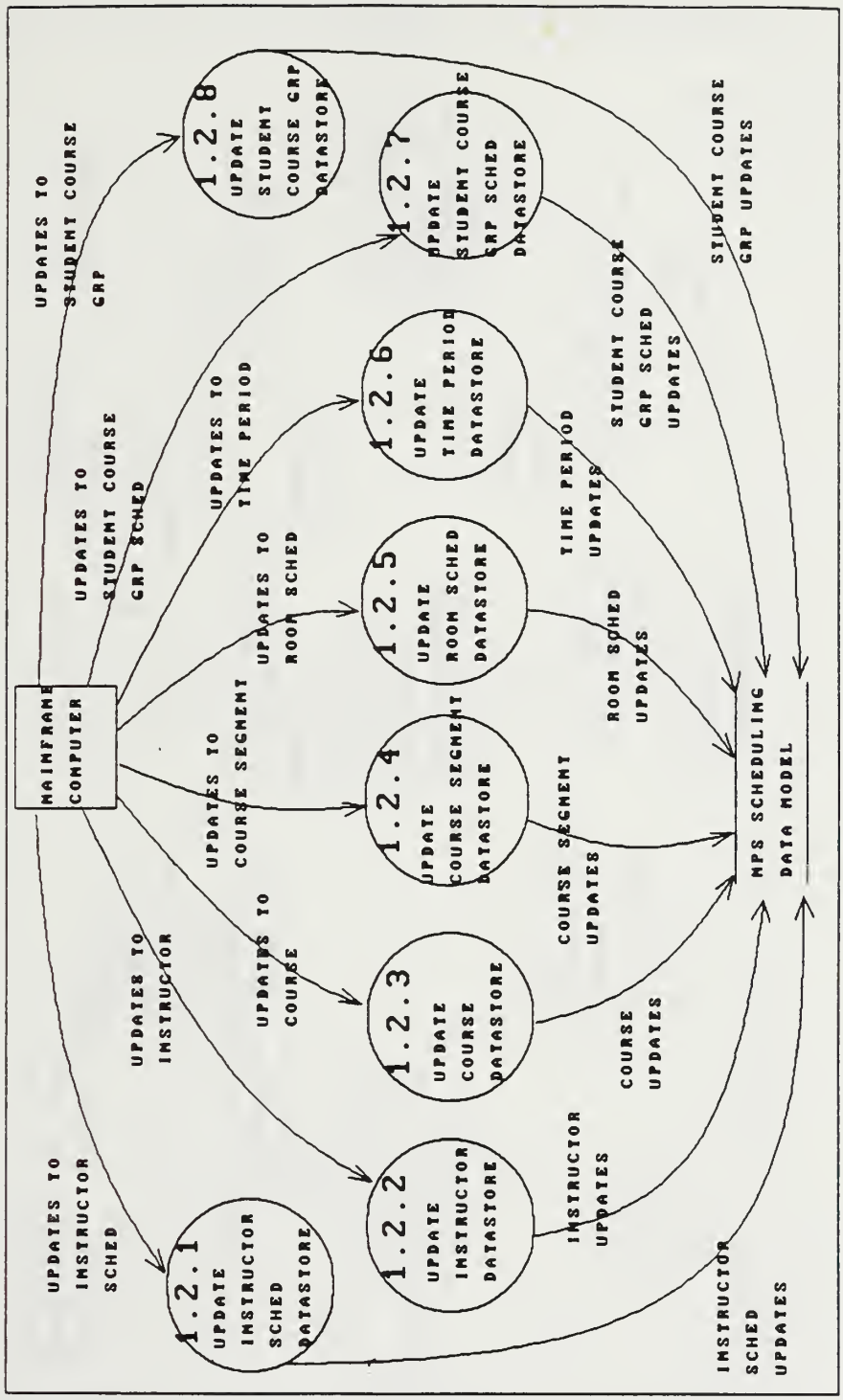

MAINFRAME COMPUTER
1.2.1 UPDATE INSTRUCTOR SCHED DATASTORE
1.2.2 UPDATE INSTRUCTOR DATASTORE
1.2.3 UPDATE COURSE DATASTORE
1.2.4 UPDATE COURSE SEGMENT DATASTORE
1.2.5 UPDATE ROOM SCHED DATASTORE
1.2.6 UPDATE TIME PERIOD DATASTORE
1.2.7 UPDATE STUDENT COURSE GRP SCHED DATASTORE
1.2.8 UPDATE STUDENT COURSE GRP DATASTORE
MPS SCHEDULING DATA MODEL
UPDATES TO INSTRUCTOR SCHED
UPDATES TO INSTRUCTOR
UPDATES TO COURSE
UPDATES TO COURSE SEGMENT
UPDATES TO ROOM SCHED
UPDATES TO TIME PERIOD
UPDATES TO STUDENT COURSE GRP SCHED
UPDATES TO STUDENT COURSE GRP
INSTRUCTOR SCHED UPDATES
INSTRUCTOR UPDATES
COURSE UPDATES
COURSE SEGMENT UPDATES
ROOM SCHED UPDATES
TIME PERIOD UPDATES
STUDENT COURSE GRP SCHED UPDATES
STUDENT COURSE GRP UPDATES

FIG. C-6, Logical Dataflow Diagram for Scheduling Data Maintenance.

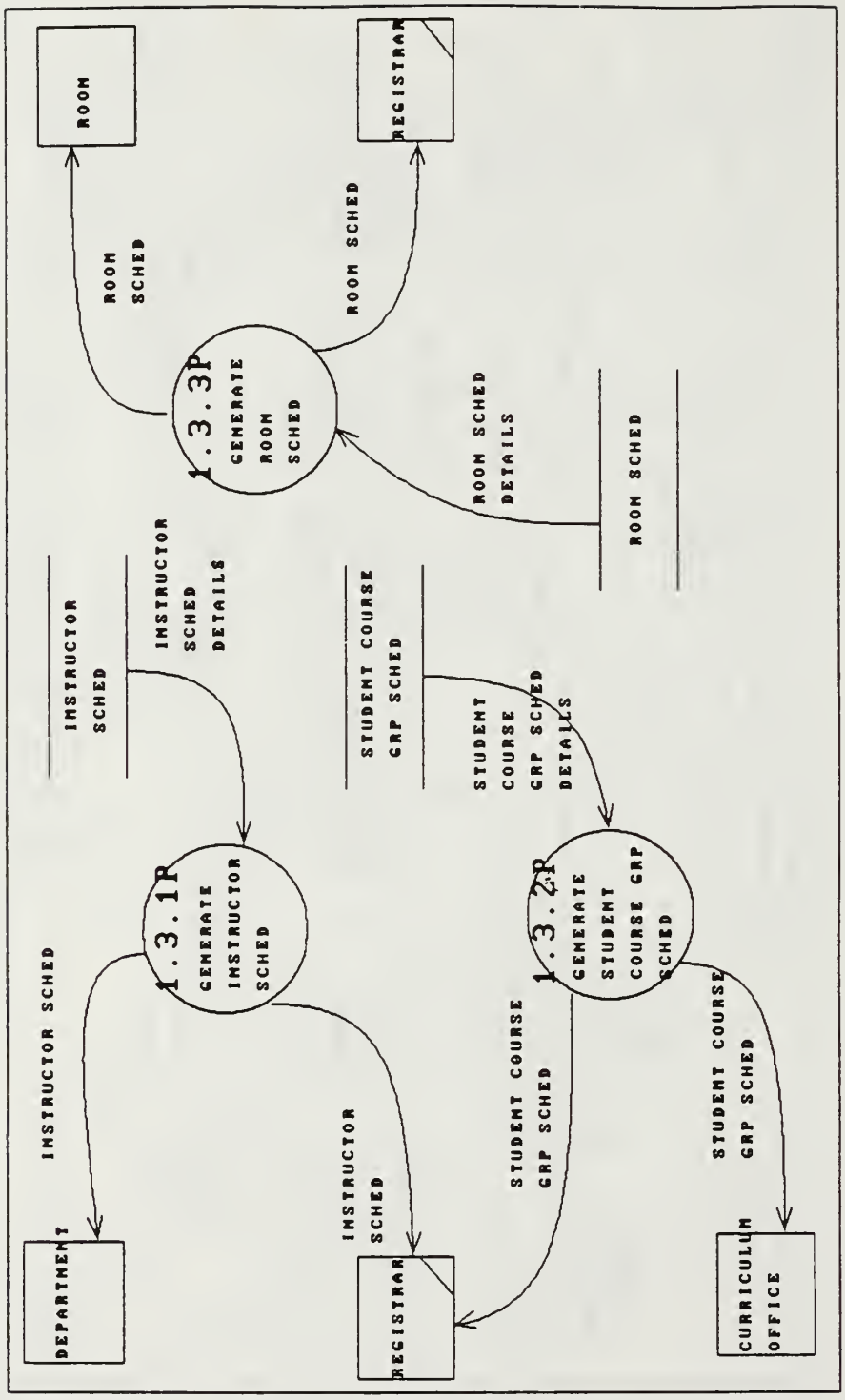


FIG. C-7, Logical Dataflow Diagram for the target Schedules and Reporting support.

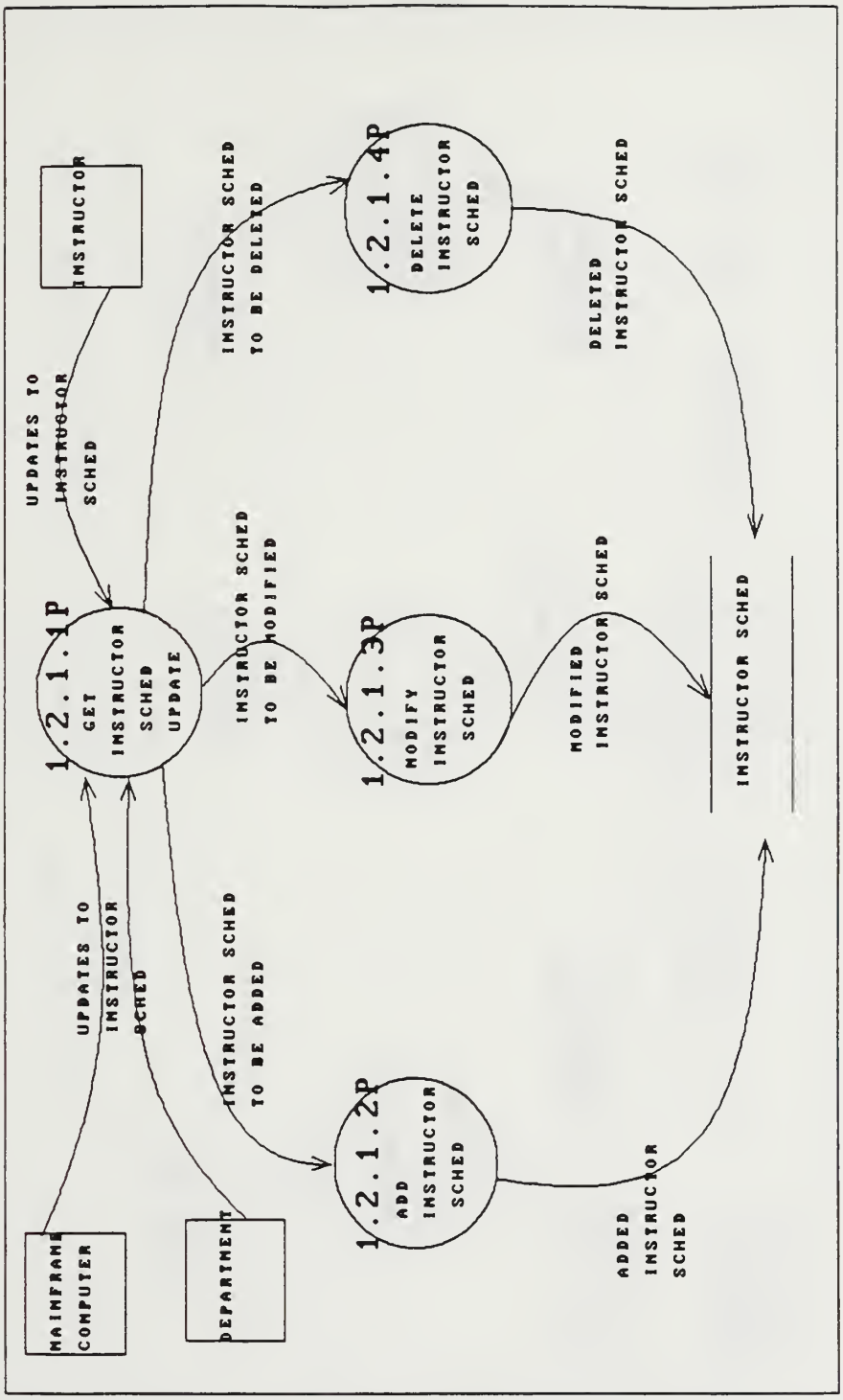


FIG. C-8, Logical Dataflow Diagram for INSTRUCTOR SCHED Data Store Maintenance.

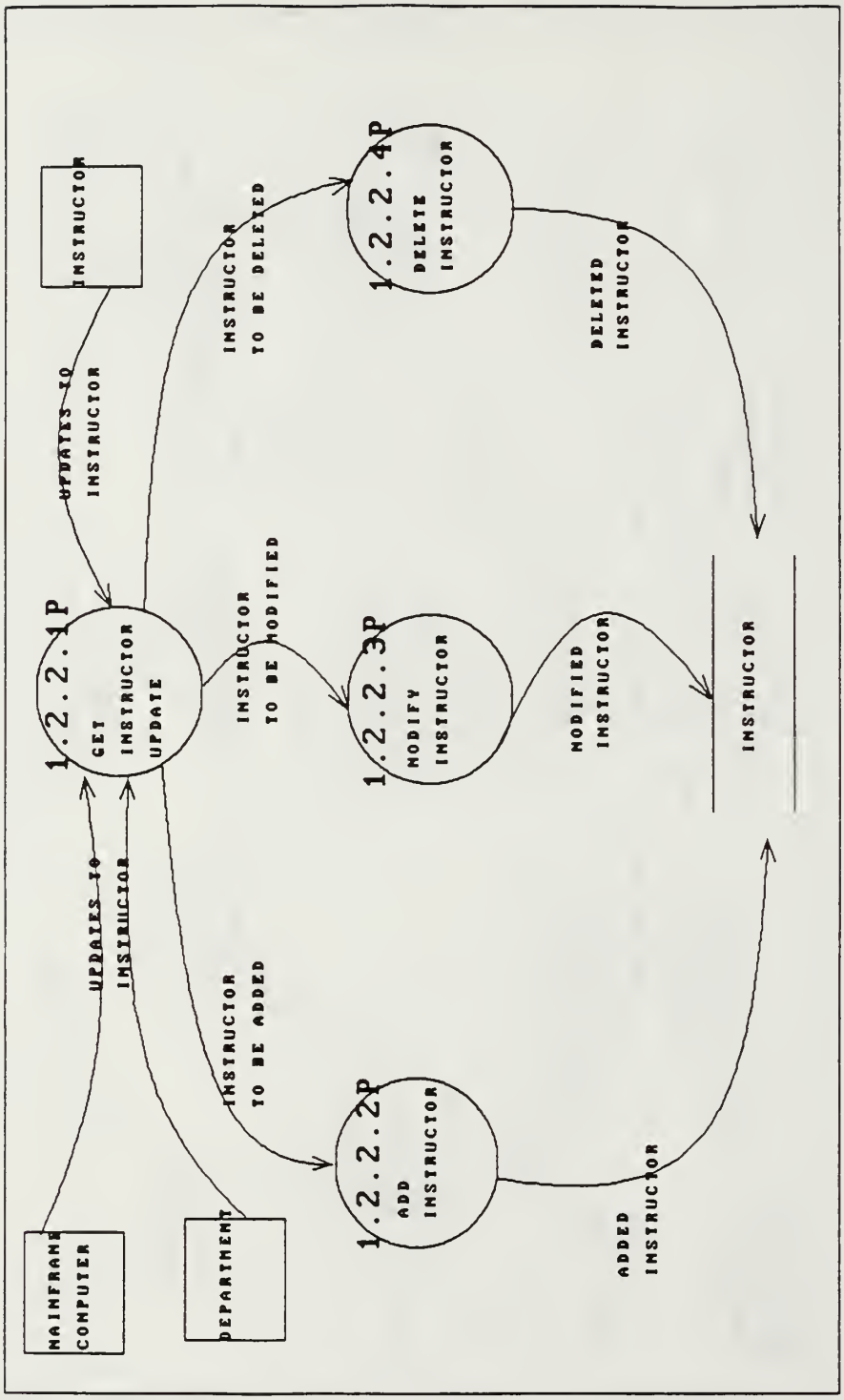


FIG. C-9, Logical Dataflow Diagram for INSTRUCTOR Data Store Maintenance.

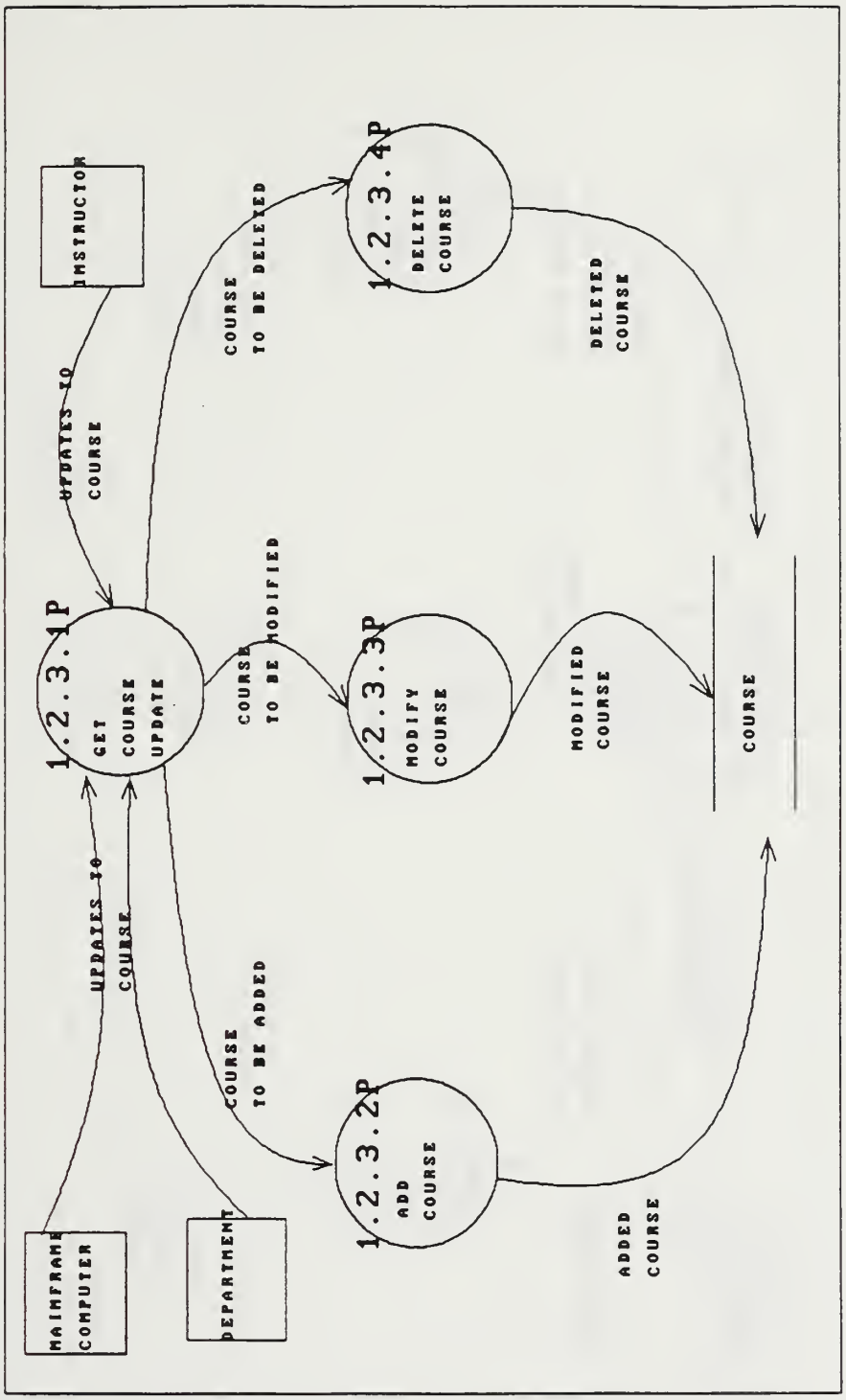


FIG. C-10, Logical Dataflow Diagram for COURSE Data Store Maintenance.

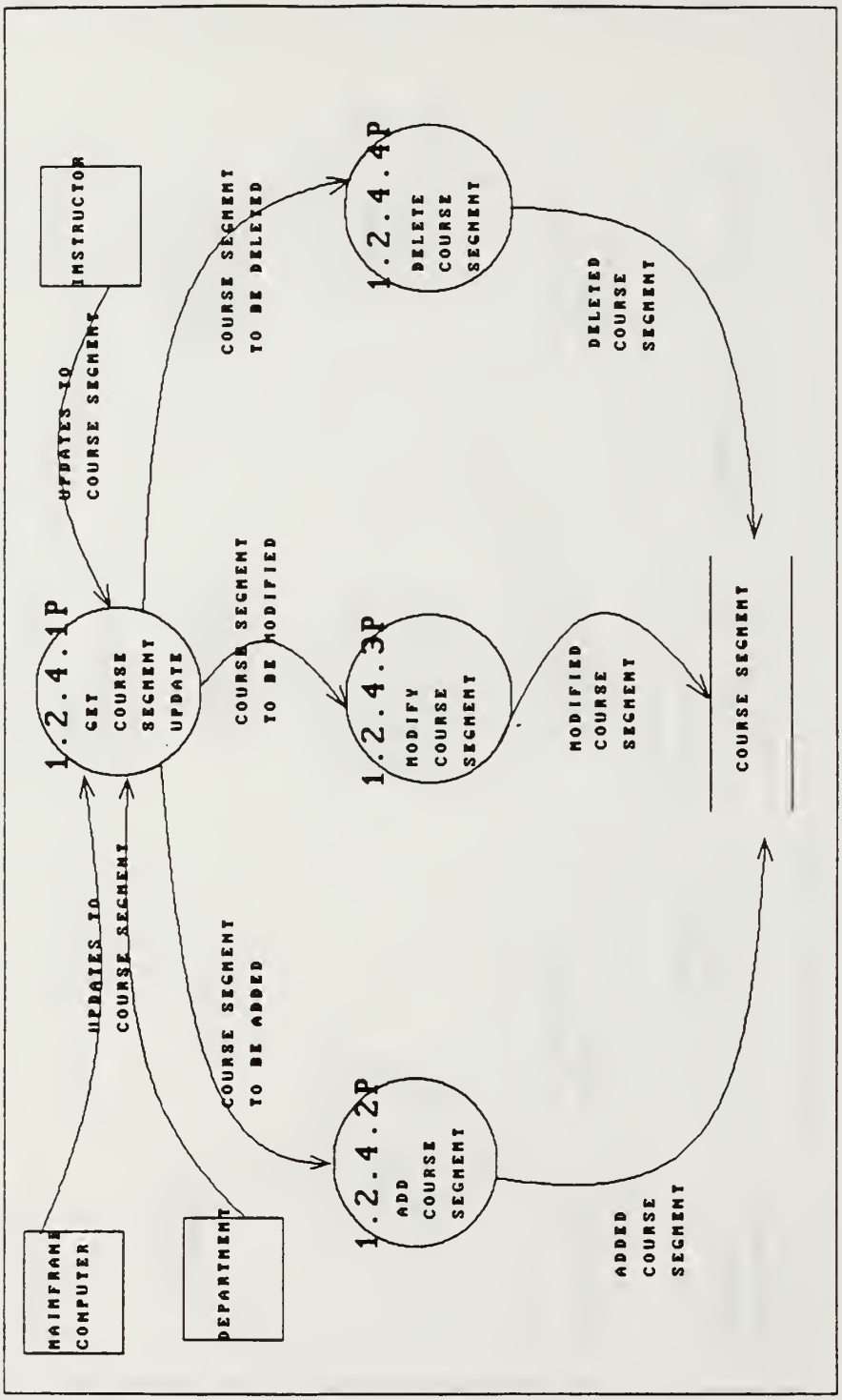


FIG. C-11, Logical Dataflow Diagram for COURSE SEGMENT Data Store Maintenance.

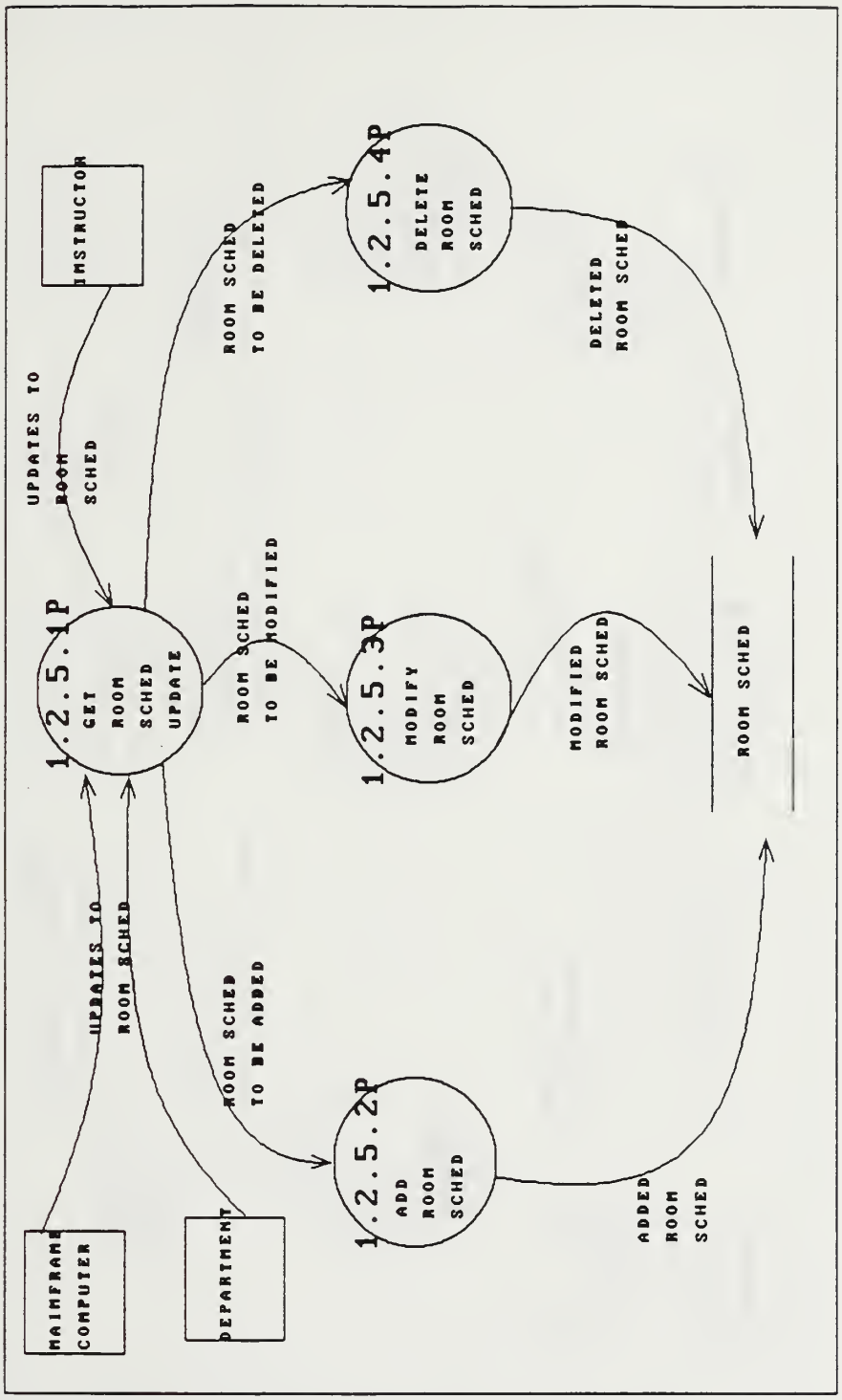


FIG. C-12, Logical Dataflow Diagram for ROOM SCHED Data Store Maintenance.

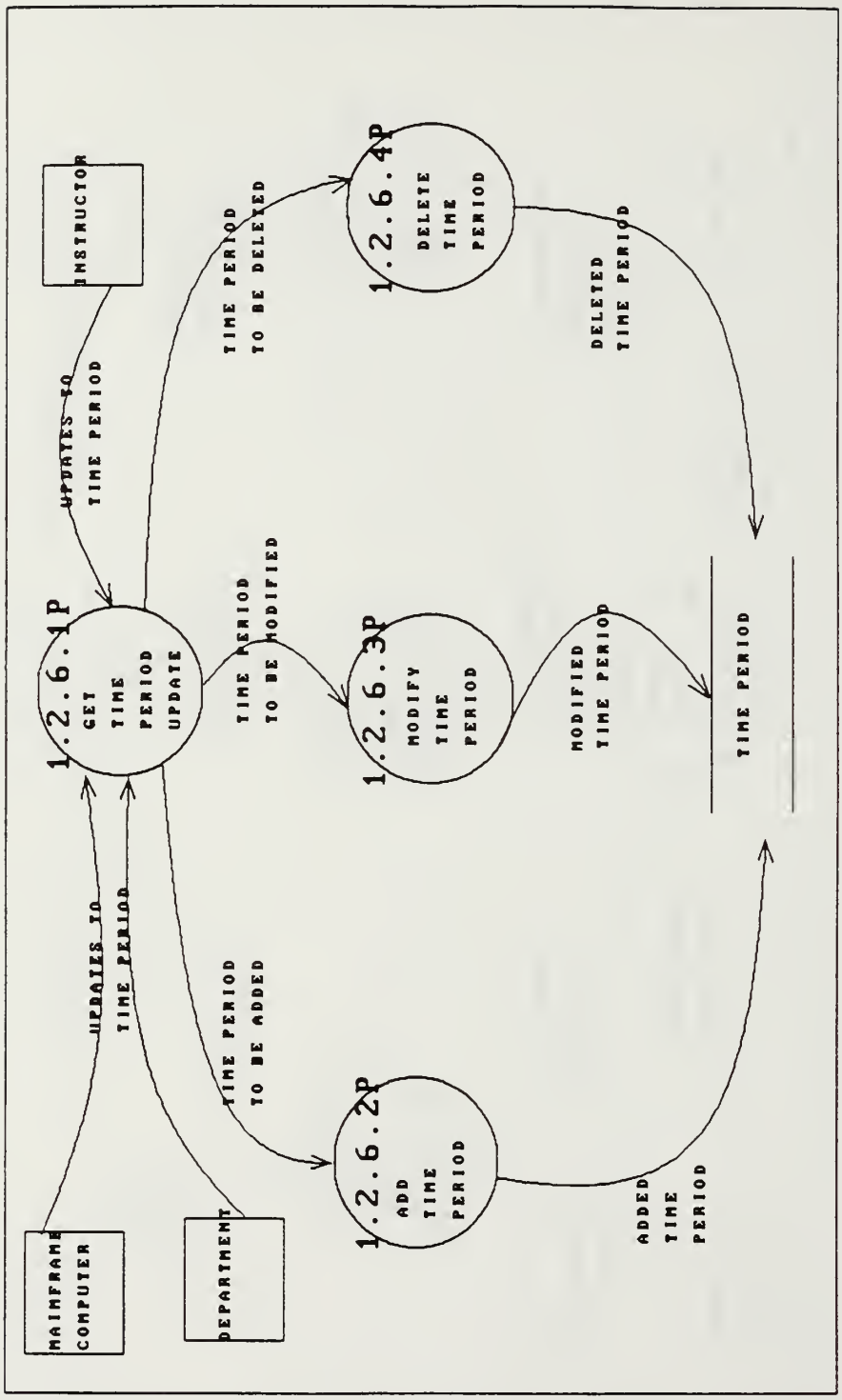


FIG. C-13, Logical Dataflow Diagram for TIME PERIOD Data Store Maintenance.

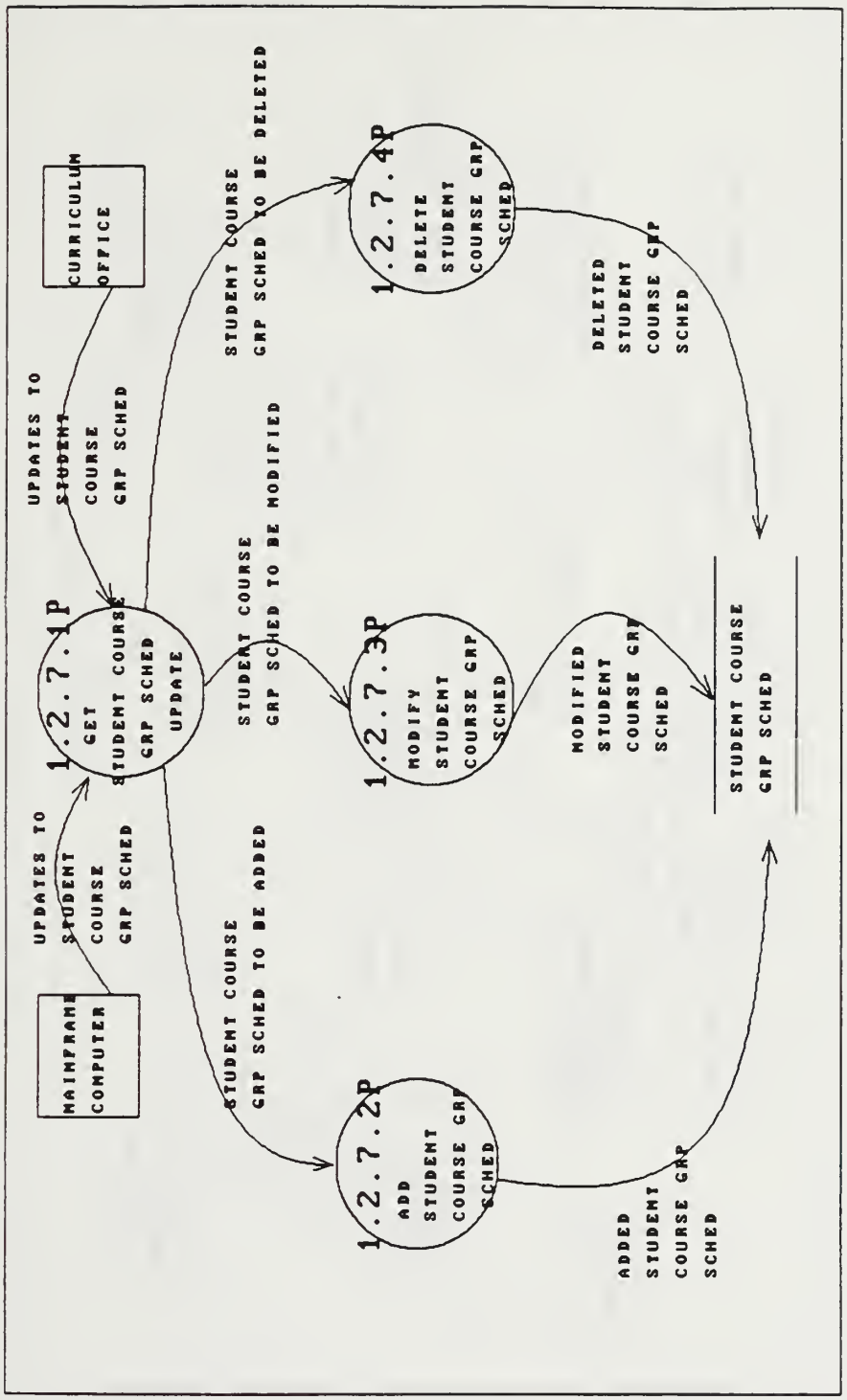

CURRICULUM OFFICE
MAINFRAME COMPUTER
UPDATES TO STUDENT COURSE GRP SCHED
UPDATES TO STUDENT COURSE GRP SCHED
1.2.7.1P GET STUDENT COURSE GRP SCHED UPDATE
STUDENT COURSE GRP SCHED TO BE ADDED
STUDENT COURSE GRP SCHED TO BE MODIFIED
STUDENT COURSE GRP SCHED TO BE DELETED
1.2.7.2P ADD STUDENT COURSE GRP SCHED
1.2.7.3P MODIFY STUDENT COURSE GRP SCHED
1.2.7.4P DELETE STUDENT COURSE GRP SCHED
ADDED STUDENT COURSE GRP SCHED
MODIFIED STUDENT COURSE GRP SCHED
DELETED STUDENT COURSE GRP SCHED
STUDENT COURSE GRP SCHED

FIG. C-14, Logical Dataflow Diagram for STUDENT COURSE GRP SCHED Data Store Maintenance.

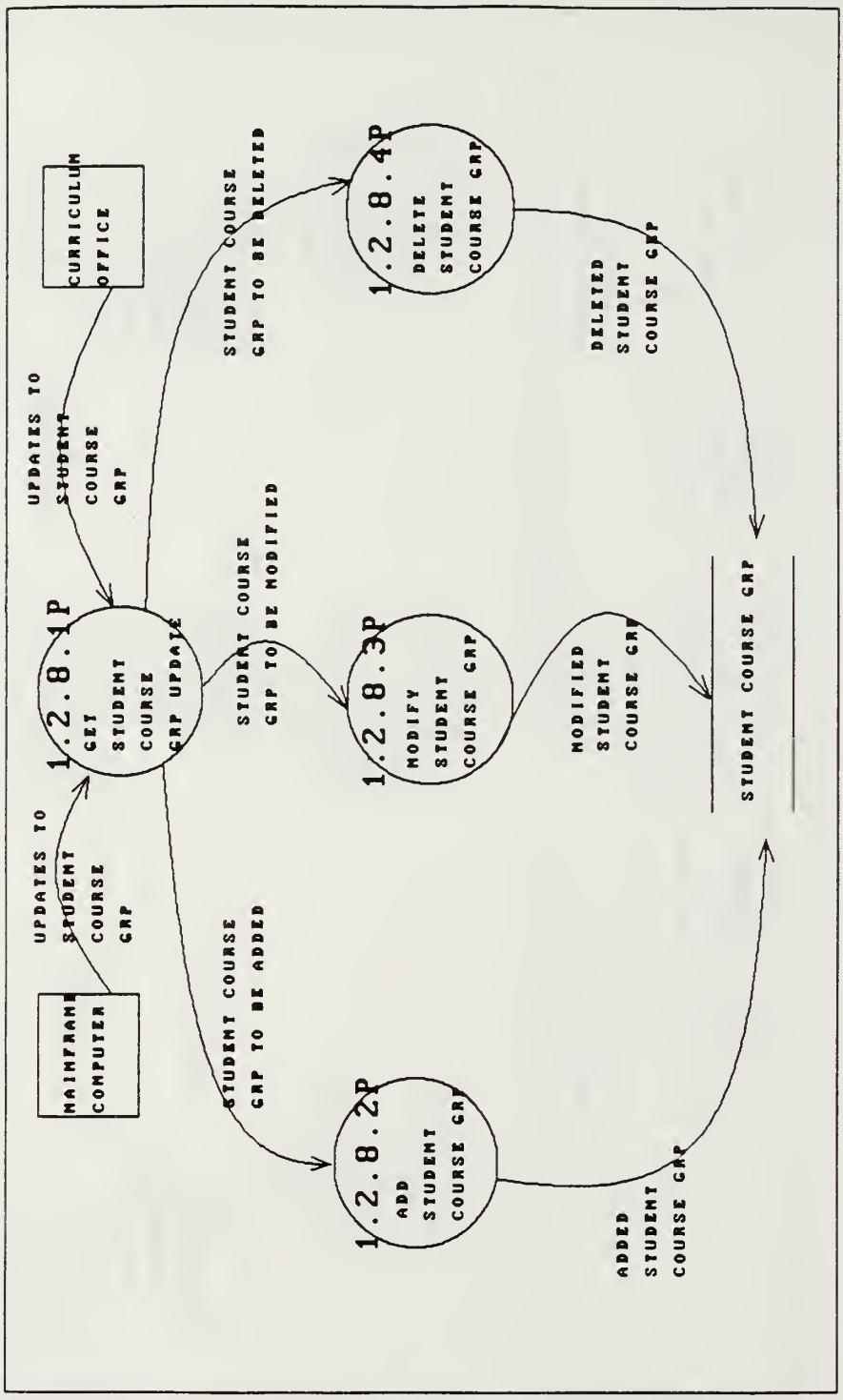

CURRICULUM OFFICE
MAINFRAME COMPUTER
UPDATES TO STUDENT COURSE GRP
UPDATES TO STUDENT COURSE GRP
1.2.8.1P GET STUDENT COURSE GRP UPDATE
STUDENT COURSE GRP TO BE DELETED
STUDENT COURSE GRP TO BE MODIFIED
STUDENT COURSE GRP TO BE ADDED
1.2.8.4P DELETE STUDENT COURSE GRP
1.2.8.3P MODIFY STUDENT COURSE GRP
1.2.8.2P ADD STUDENT COURSE GRP
DELETED STUDENT COURSE GRP
MODIFIED STUDENT COURSE GRP
ADDED STUDENT COURSE GRP
STUDENT COURSE GRP

FIG. C-15, Logical Dataflow Diagram for STUDENT COURSE GRP Data Store Maintenance.

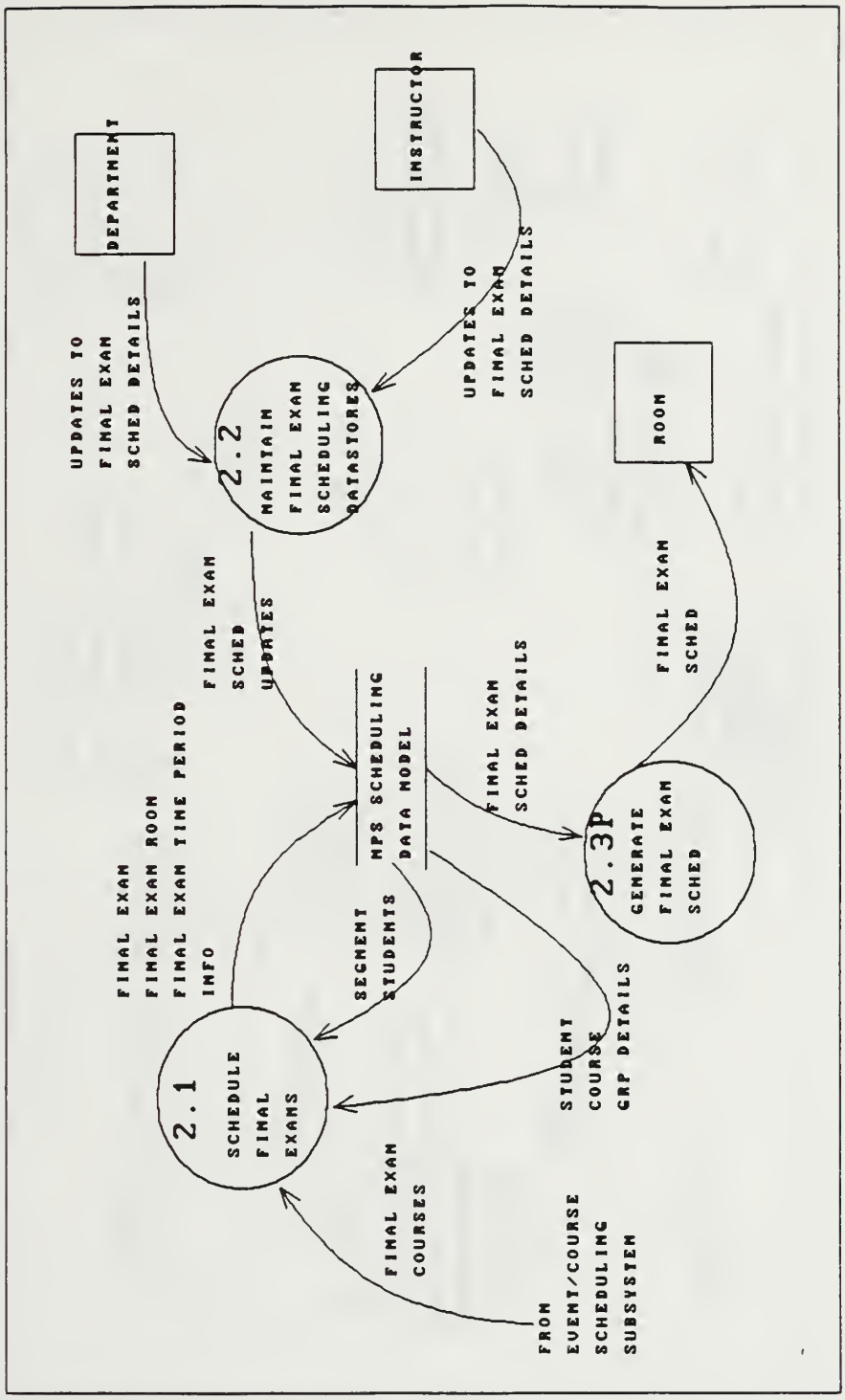


FIG. C-16, Logical Dataflow Diagram for the target Final Exam Scheduling Subsystem.

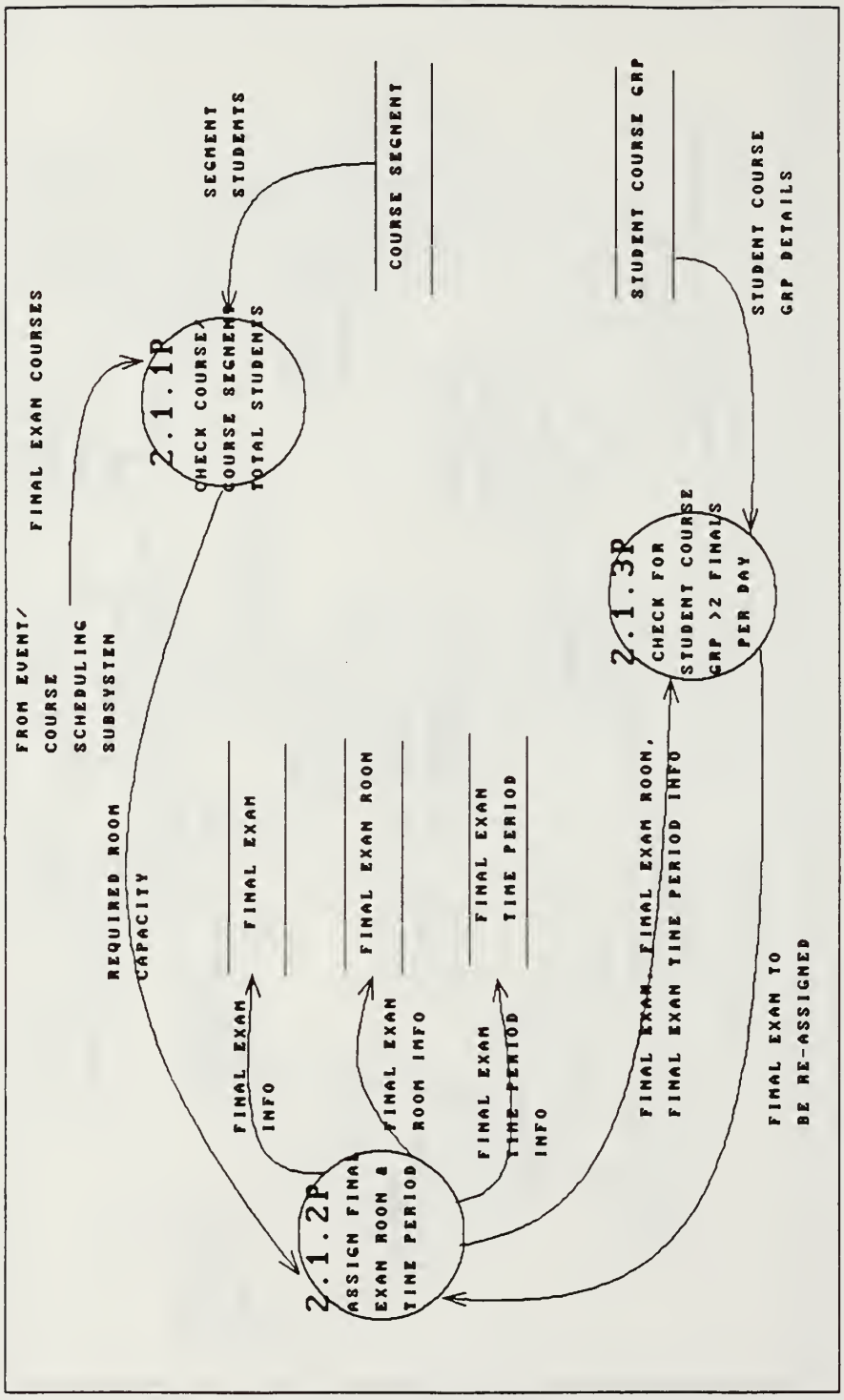

FROM EVENT/ COURSE SCHEDULING SUBSYSTEM
FINAL EXAM COURSES
REQUIRED ROOM CAPACITY
2.1.1P
CHECK COURSE/ COURSE SEGMENT TOTAL STUDENTS
SEGMENT STUDENTS
COURSE SEGMENT
STUDENT COURSE GRP
STUDENT COURSE GRP DETAILS
2.1.3P
CHECK FOR STUDENT COURSE GRP >2 FINALS PER DAY
FINAL EXAM INFO
FINAL EXAM
FINAL EXAM ROOM INFO
FINAL EXAM ROOM
FINAL EXAM TIME PERIOD INFO
FINAL EXAM TIME PERIOD
2.1.2P
ASSIGN FINAL EXAM ROOM & TIME PERIOD
FINAL EXAM, FINAL EXAM ROOM, FINAL EXAM TIME PERIOD INFO
FINAL EXAM TO BE RE-ASSIGNED

FIG. C-17, Logical Dataflow Diagram for the target Final Exam Scheduling.

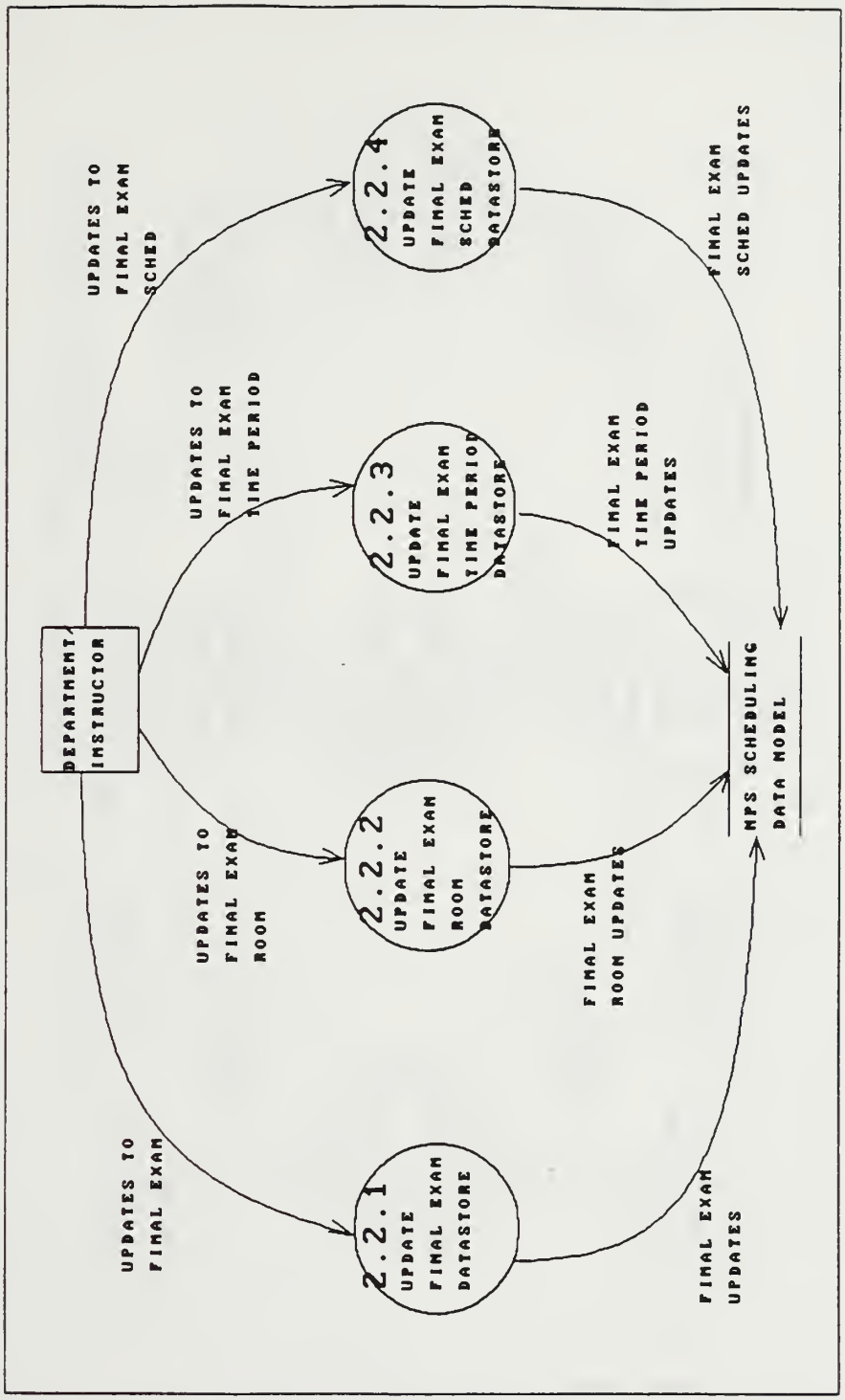

DEPARTMENT INSTRUCTOR
UPDATES TO FINAL EXAM SCHED
UPDATES TO FINAL EXAM TIME PERIOD
UPDATES TO FINAL EXAM ROOM
UPDATES TO FINAL EXAM
2.2.4 UPDATE FINAL EXAM SCHED DATASTORE
2.2.3 UPDATE FINAL EXAM TIME PERIOD DATASTORE
2.2.2 UPDATE FINAL EXAM ROOM DATASTORE
2.2.1 UPDATE FINAL EXAM DATASTORE
FINAL EXAM SCHED UPDATES
FINAL EXAM TIME PERIOD UPDATES
FINAL EXAM ROOM UPDATES
FINAL EXAM UPDATES
MPS SCHEDULING DATA MODEL

FIG. C-18, Logical Dataflow Diagram for the target Final Exam Data Maintenance.

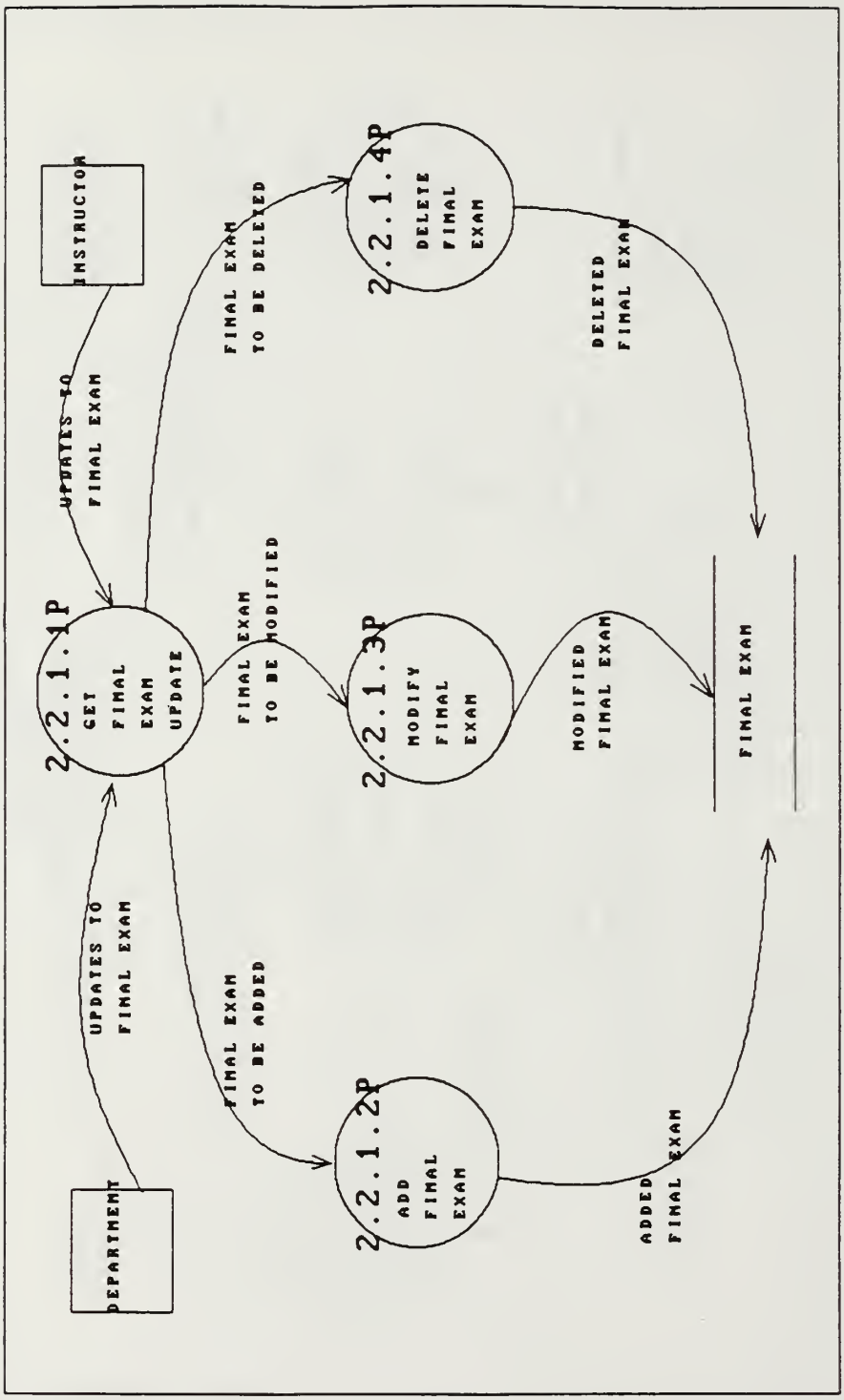


FIG. C-19, Logical Dataflow Diagram for FINAL EXAM Data Store Maintenance.

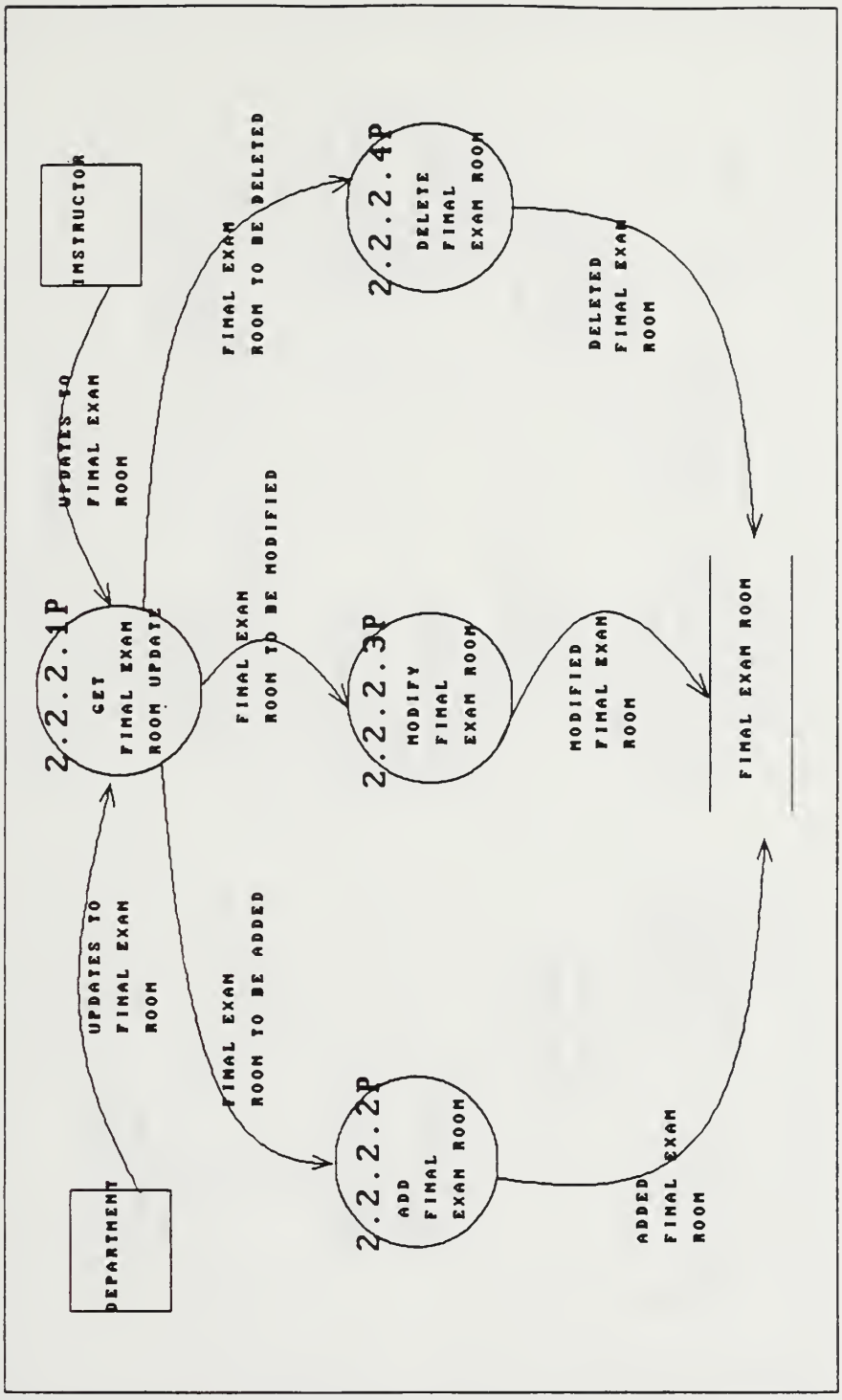

INSTRUCTOR
UPDATES TO FINAL EXAM ROOM
2.2.2.1P GET FINAL EXAM ROOM UPDATE
FINAL EXAM ROOM TO BE DELETED
2.2.2.4P DELETE FINAL EXAM ROOM
DELETED FINAL EXAM ROOM
FINAL EXAM ROOM TO BE MODIFIED
2.2.2.3P MODIFY FINAL EXAM ROOM
MODIFIED FINAL EXAM ROOM
FINAL EXAM ROOM
DEPARTMENT
UPDATES TO FINAL EXAM ROOM
FINAL EXAM ROOM TO BE ADDED
2.2.2.2P ADD FINAL EXAM ROOM
ADDED FINAL EXAM ROOM

FIG. C-20, Logical Dataflow Diagram for FINAL EXAM ROOM Data Store Maintenance.

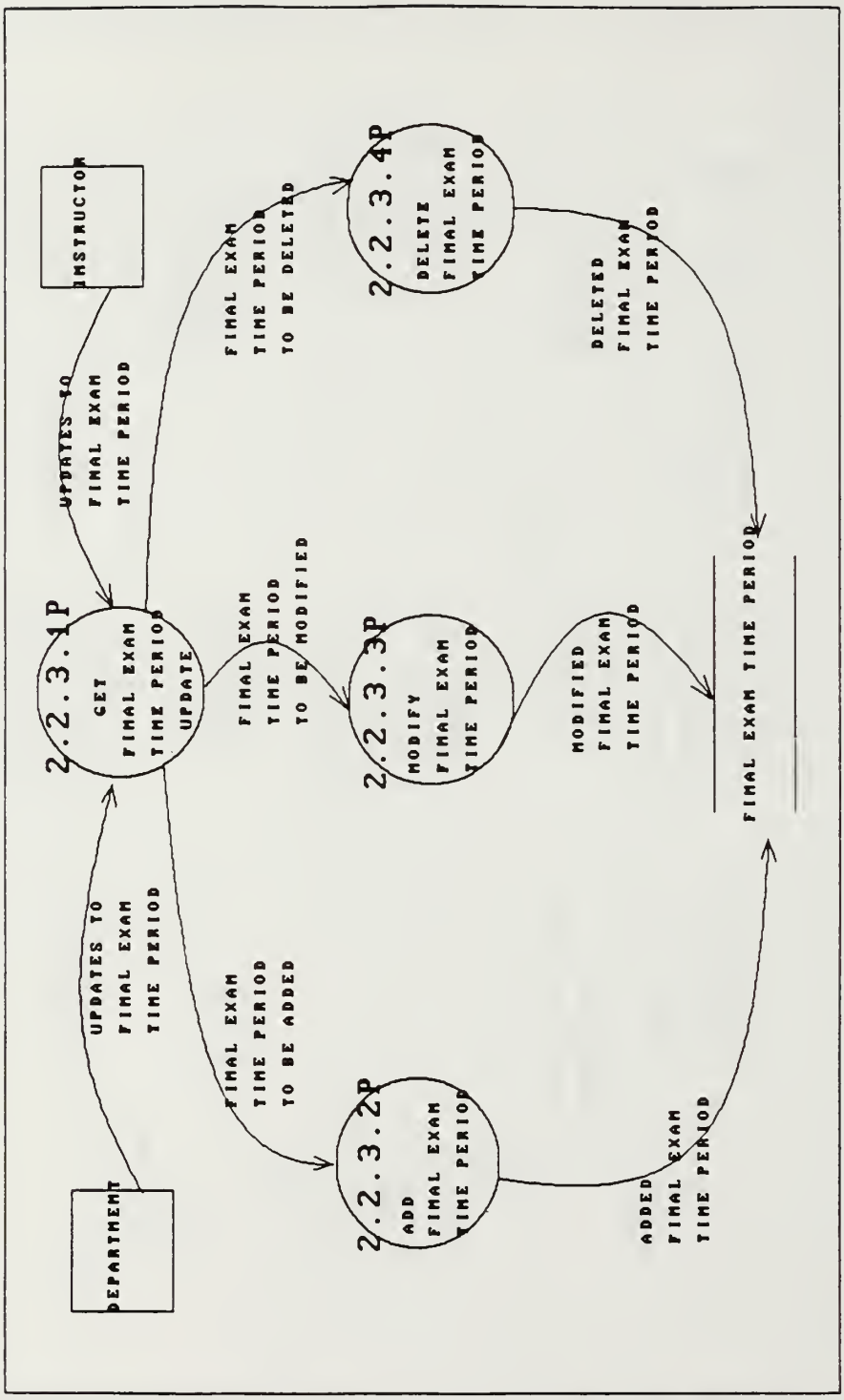


FIG. C-21, Logical Dataflow Diagram for FINAL EXAM TIME PERIOD Data Store Maintenance.

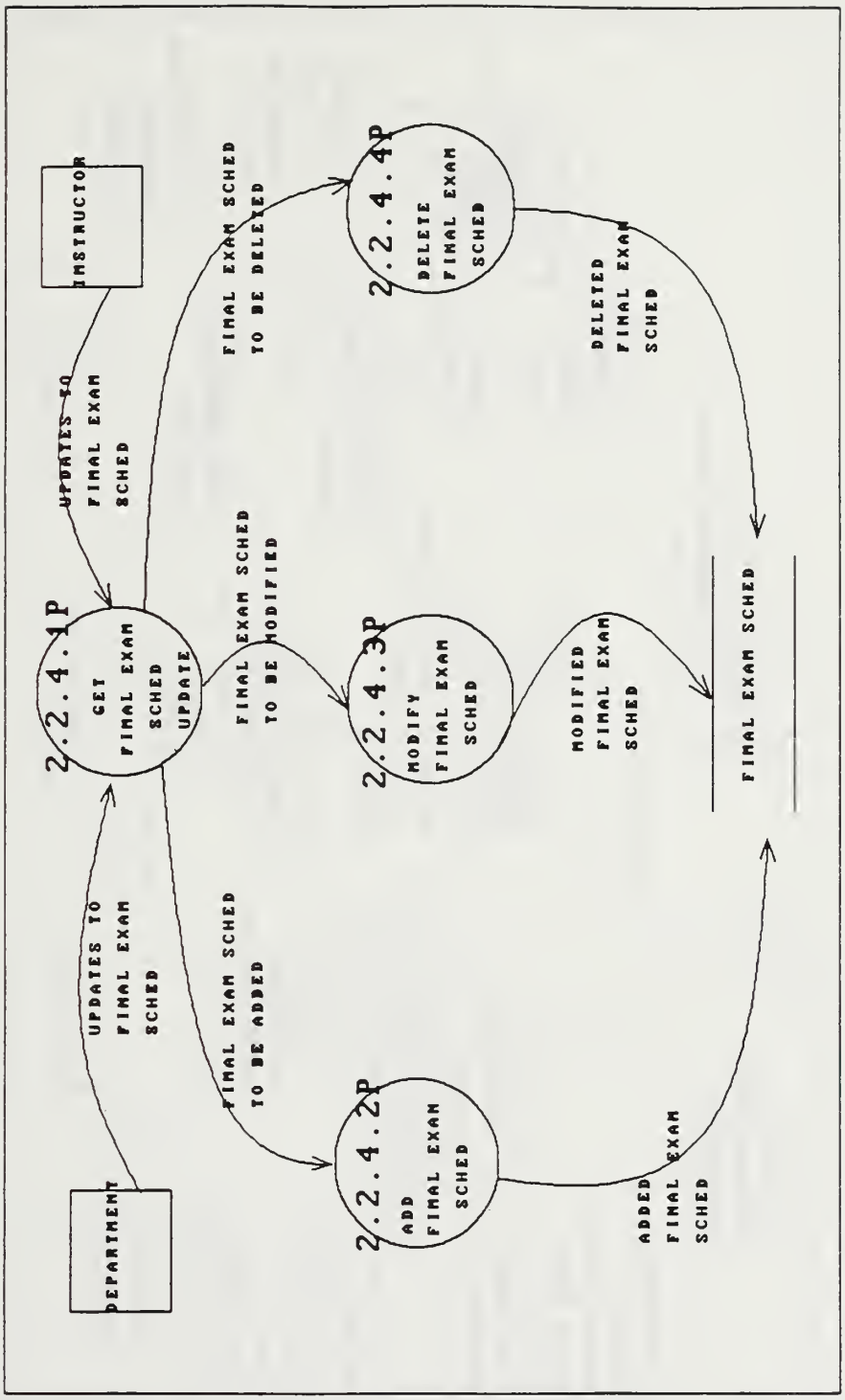


FIG. C-22, Logical Dataflow Diagram for FINAL EXAM SCHED Data Store Maintenance.

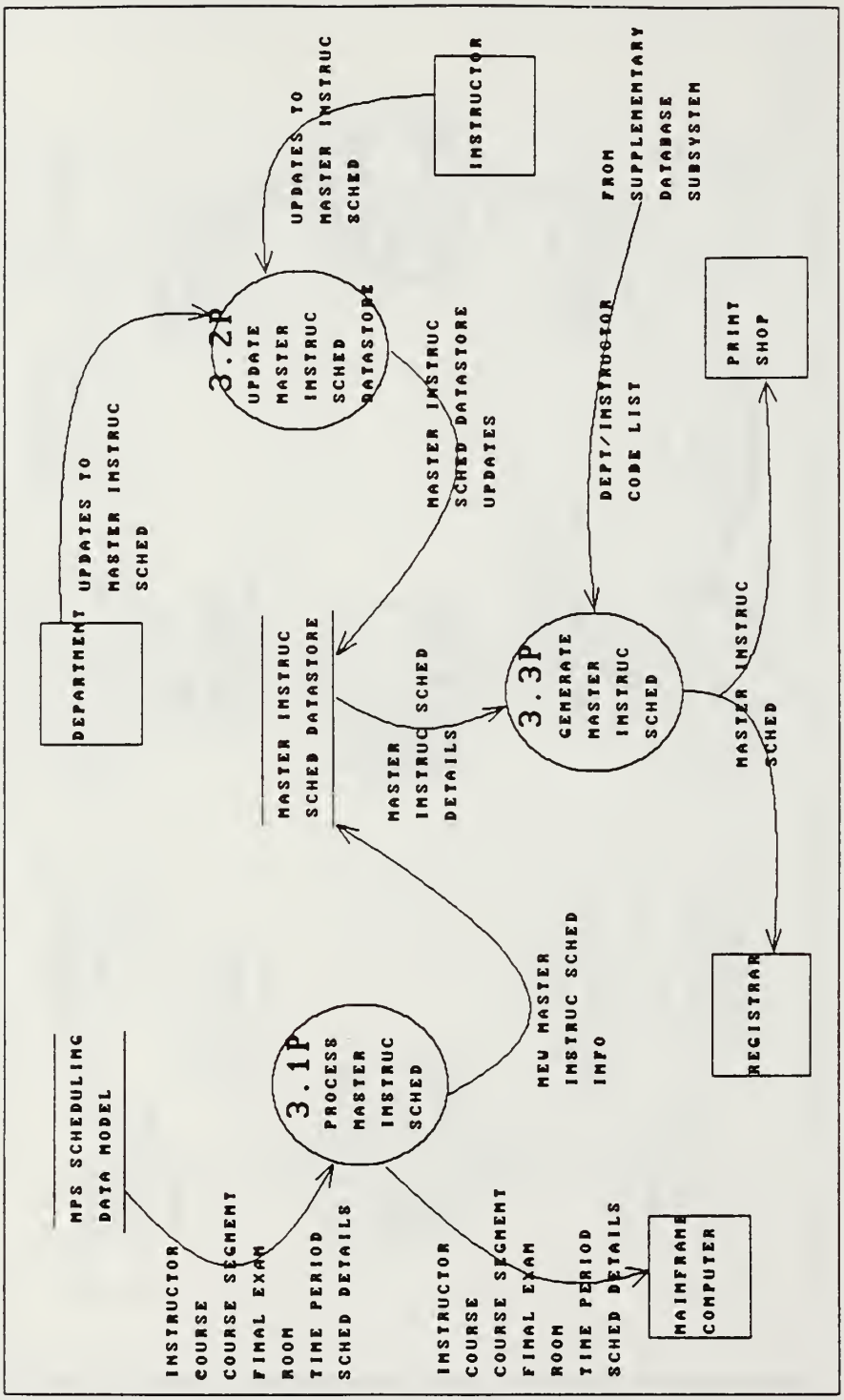


FIG. C-23, Logical Dataflow Diagram for the target Master Instruction Schedule Subsystem.

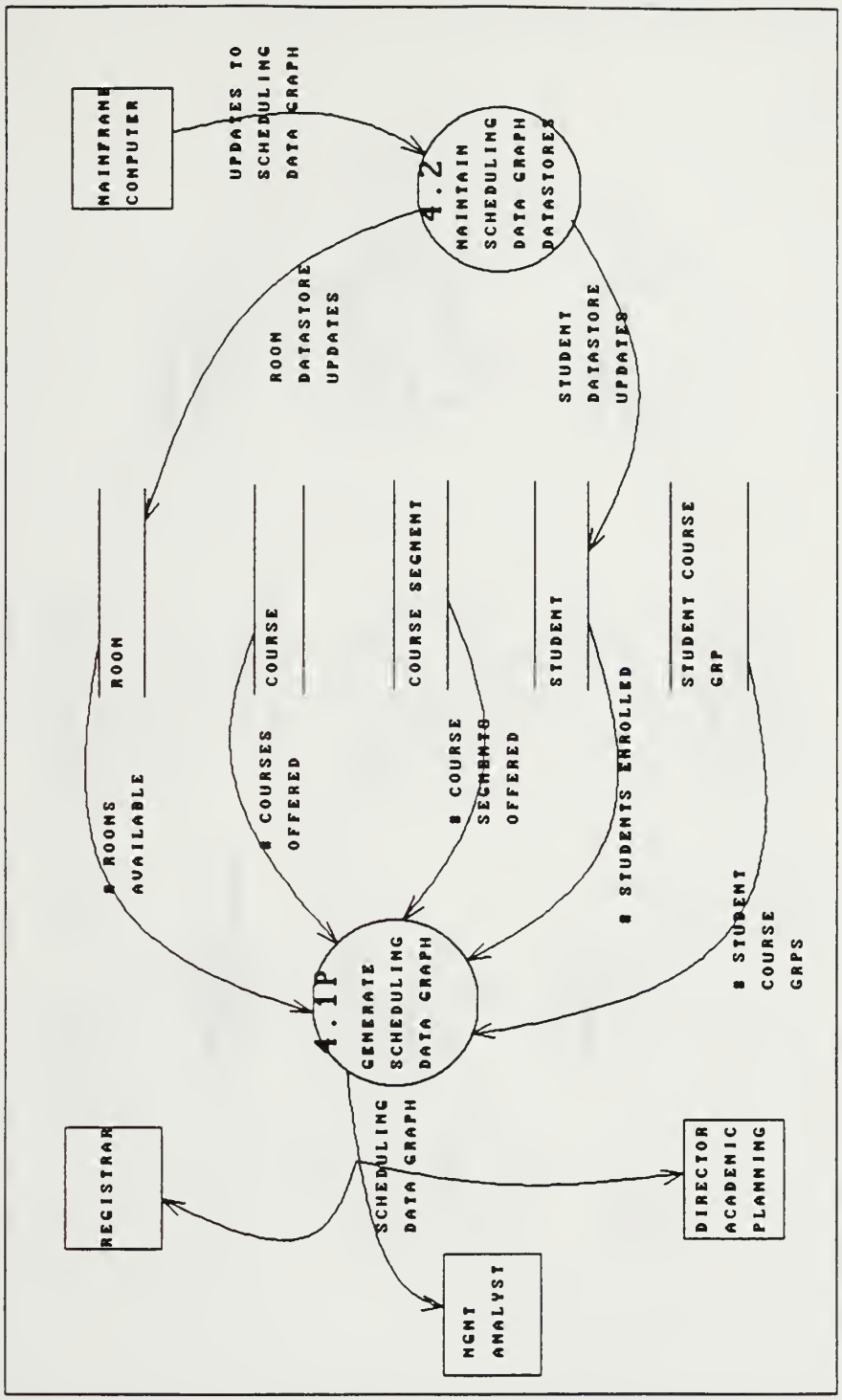


FIG. C-24, Logical Dataflow Diagram for the target Scheduling Data Graph Subsystem.

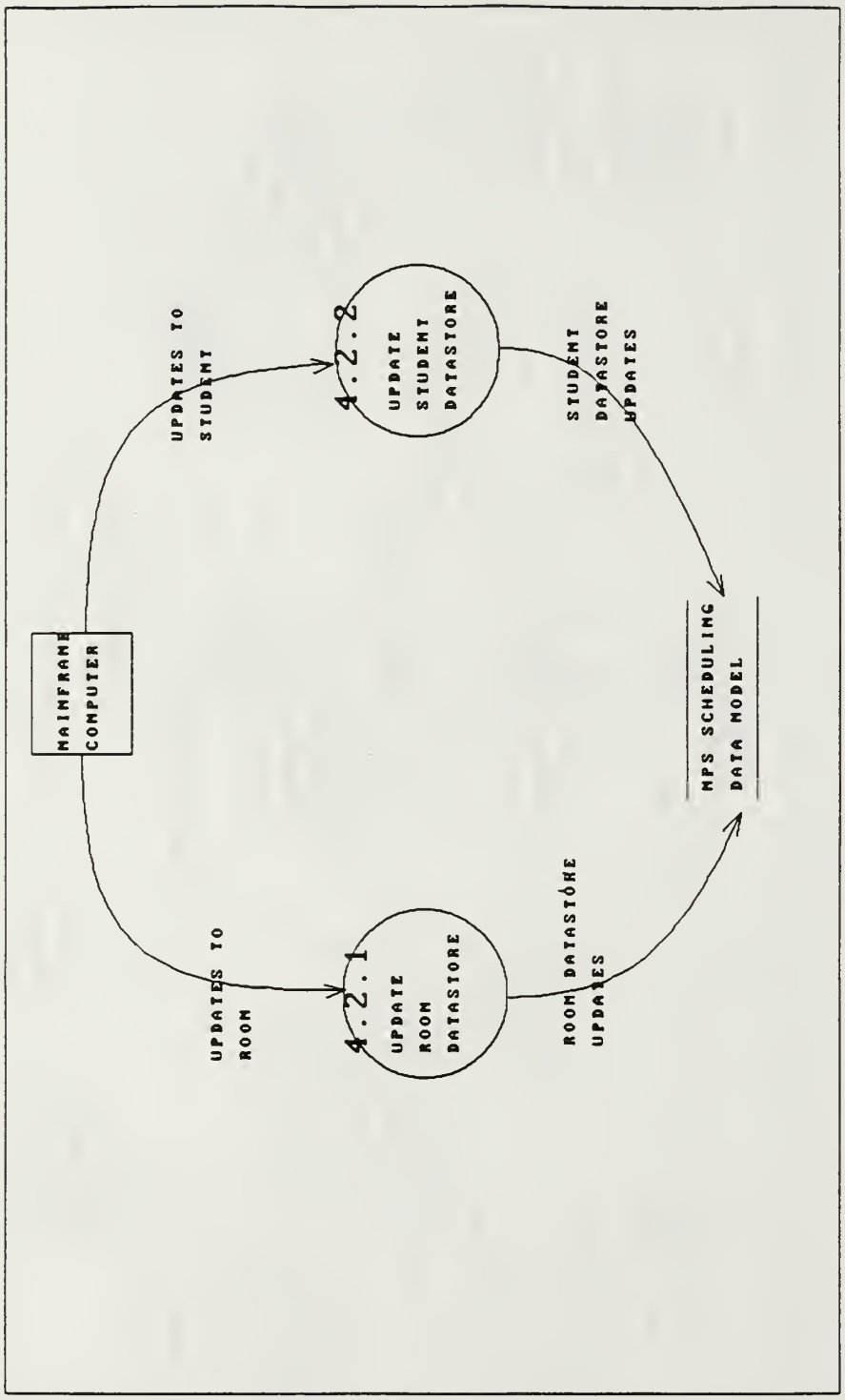


FIG. C-25, Logical Dataflow Diagram for Scheduling Data Graph Data Maintenance.

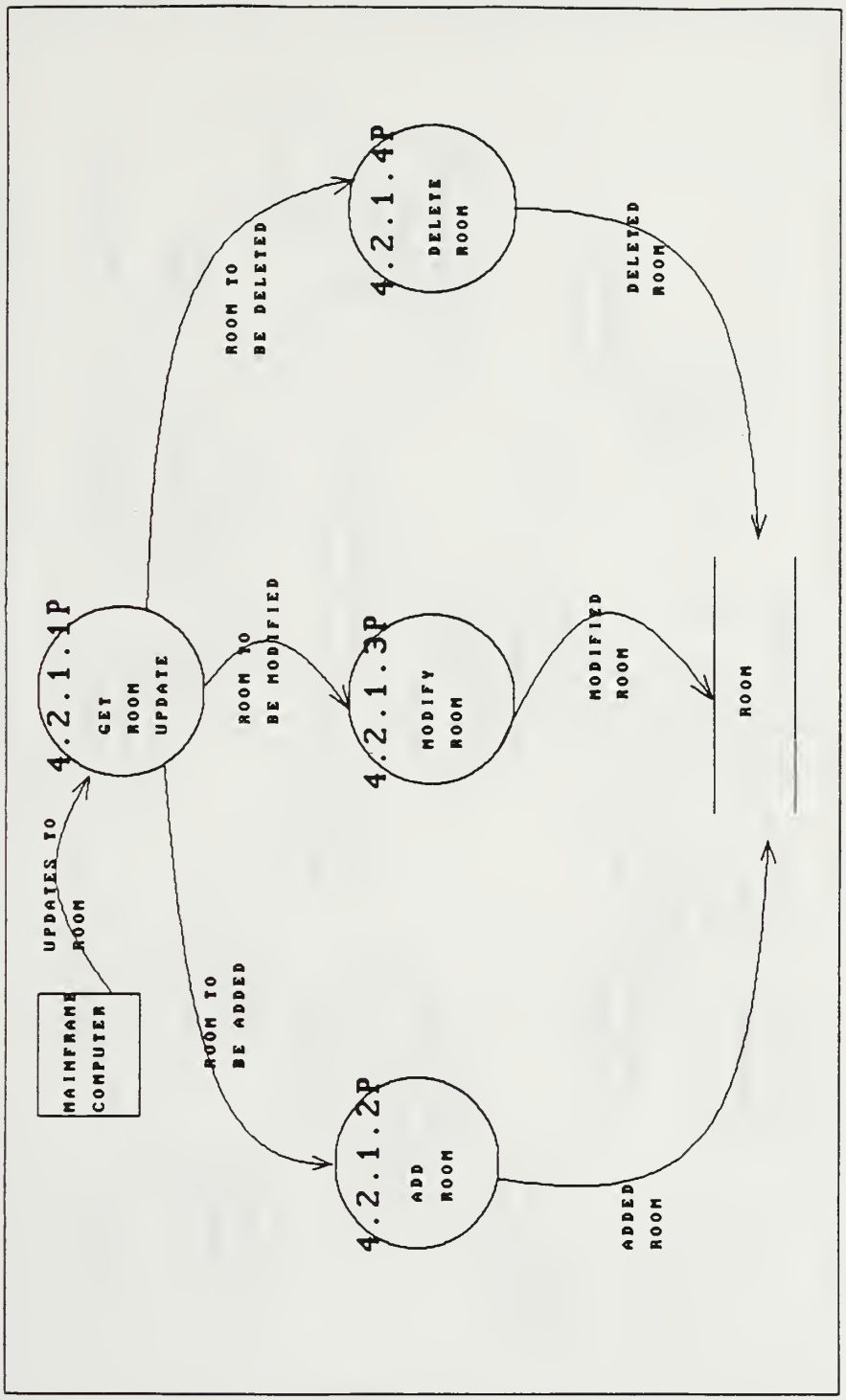


FIG. C-26, Logical Dataflow Diagram for ROOM Data Store Maintenance.

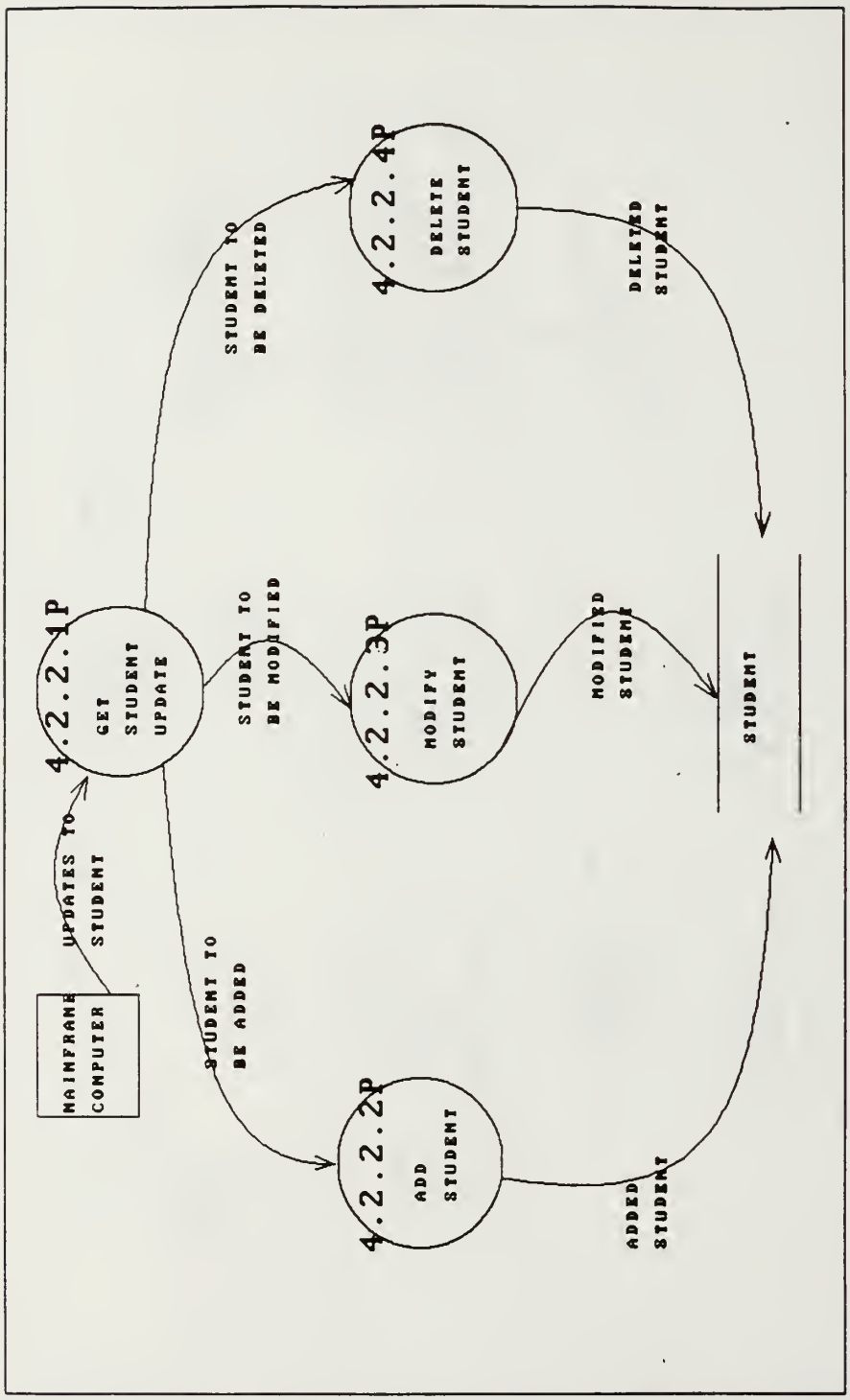


FIG. C-27, Logical Dataflow Diagram for STUDENT Data Store Maintenance.

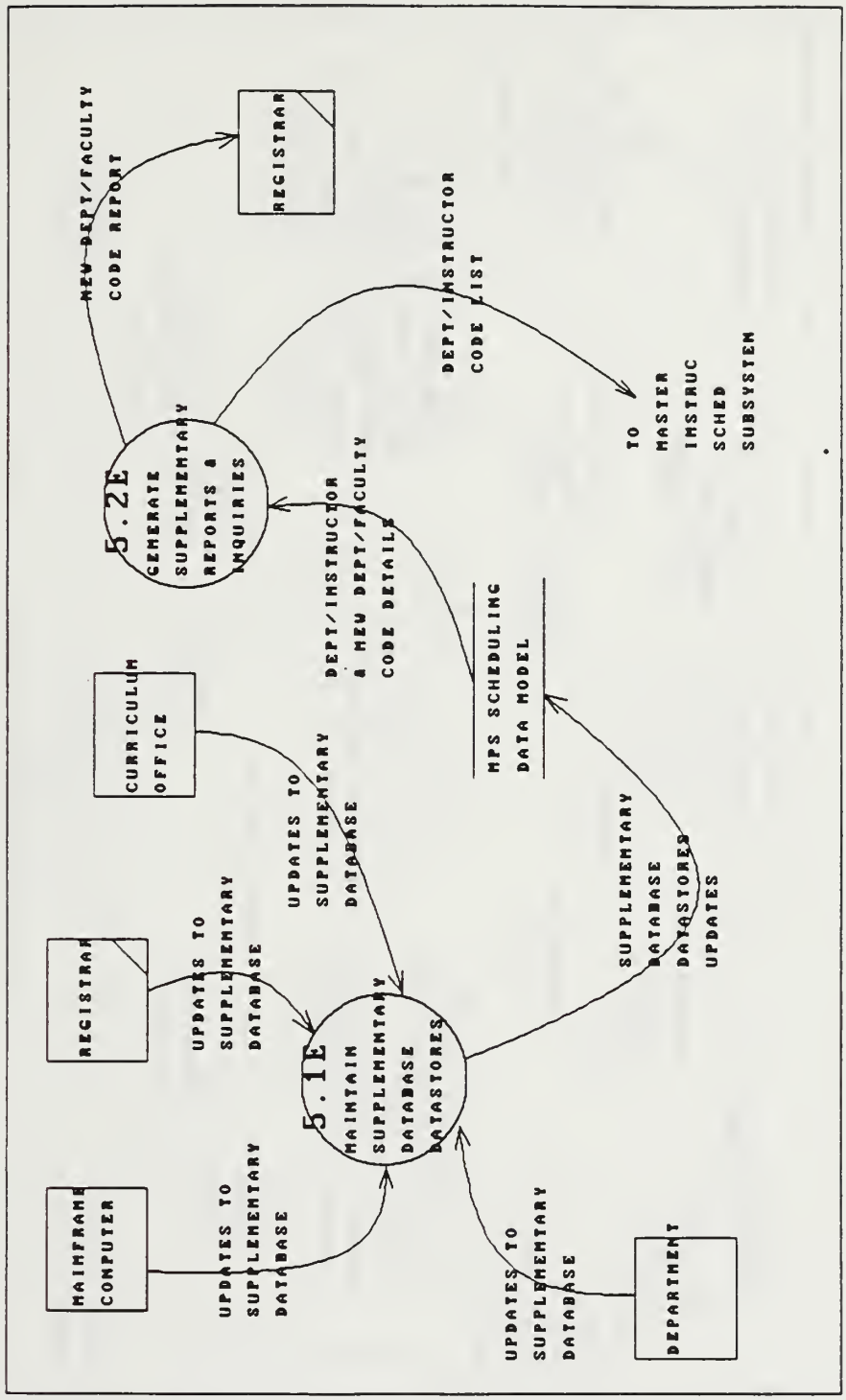


FIG. C-28, Logical Dataflow Diagram for the target Supplementary Database Subsystem.

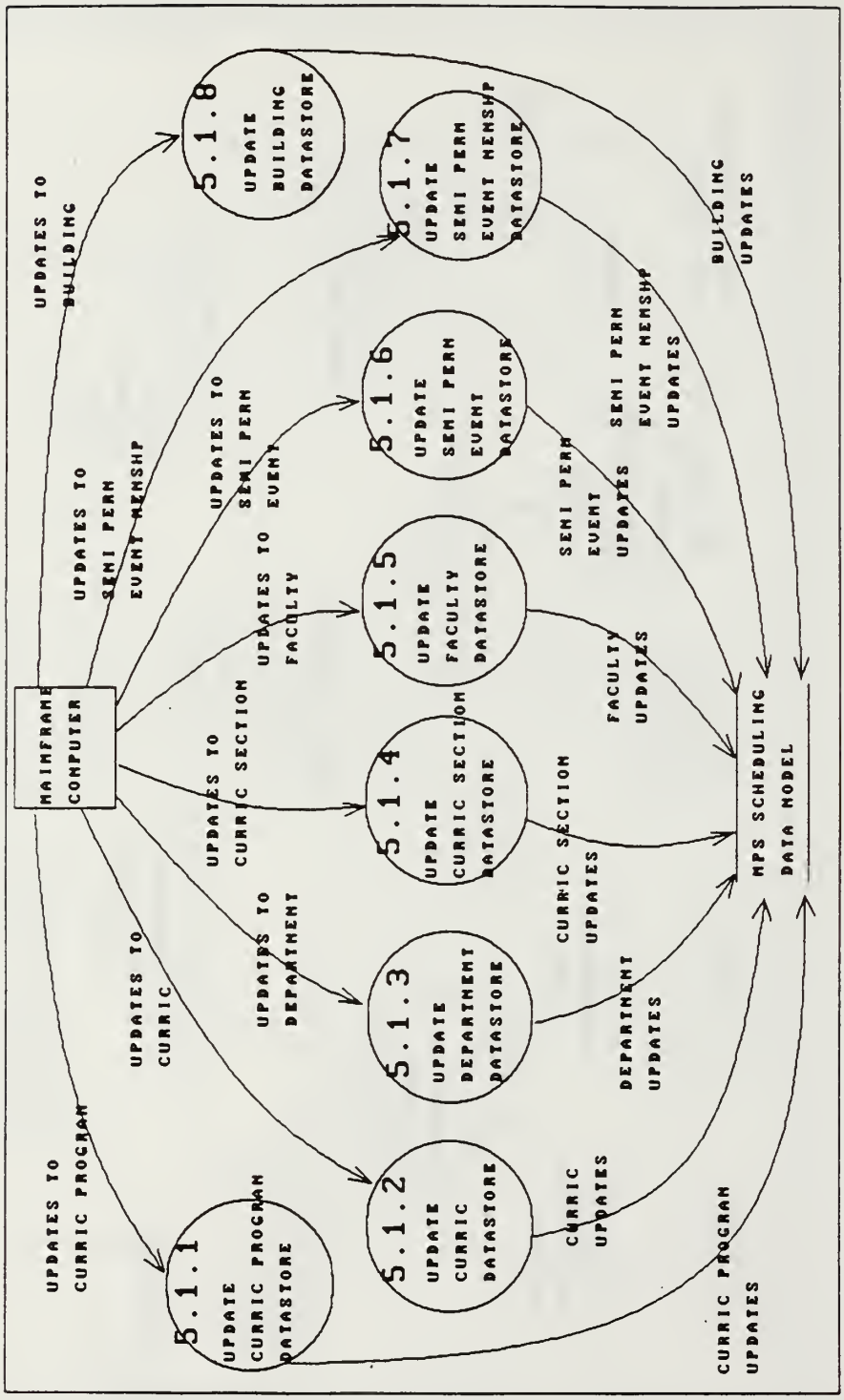


FIG. C-29, Logical Dataflow Diagram for Supplementary Database Data Maintenance.

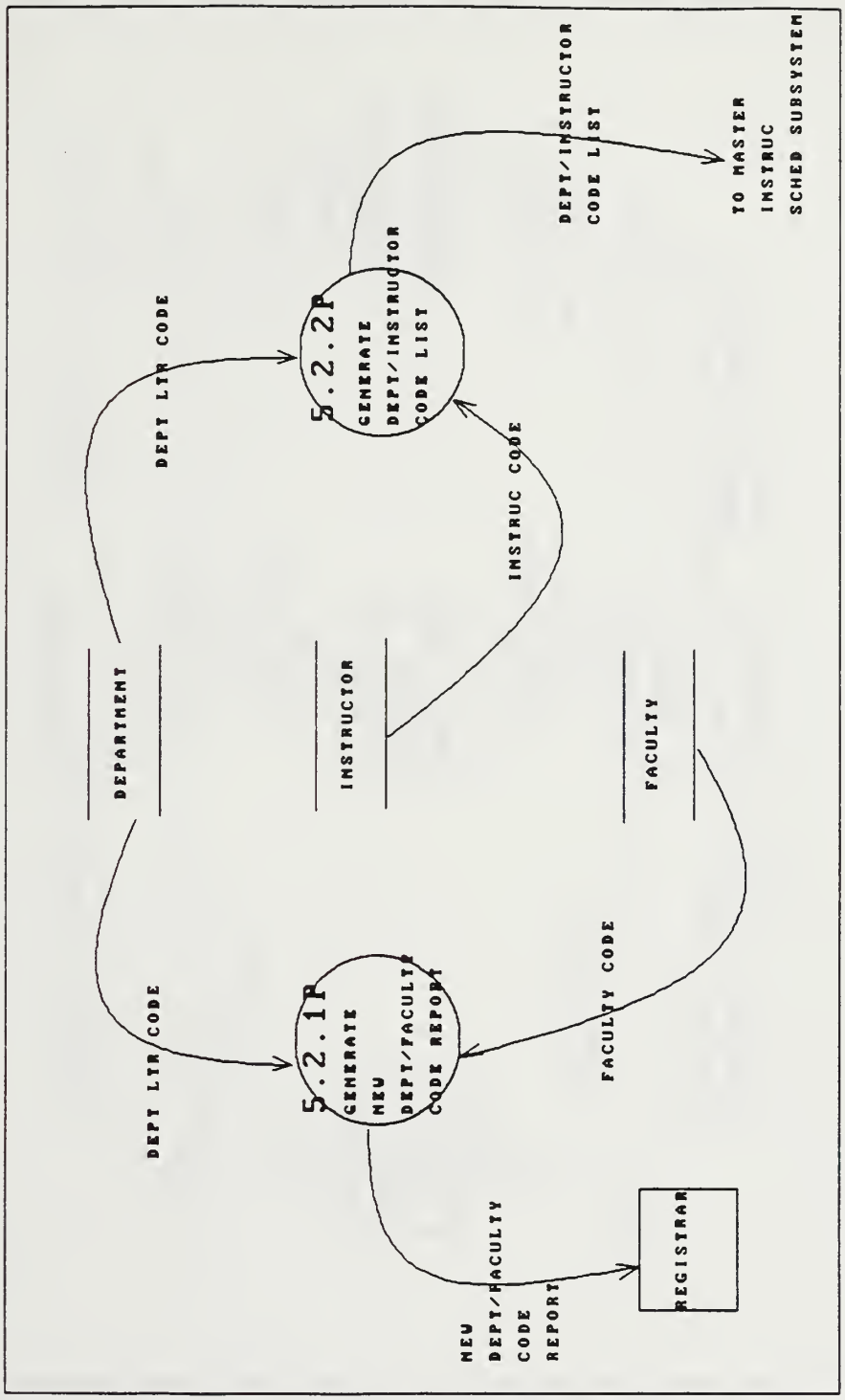


FIG. C-30, Logical Dataflow Diagram for the target Supplemetary Database Reports and Lists support.

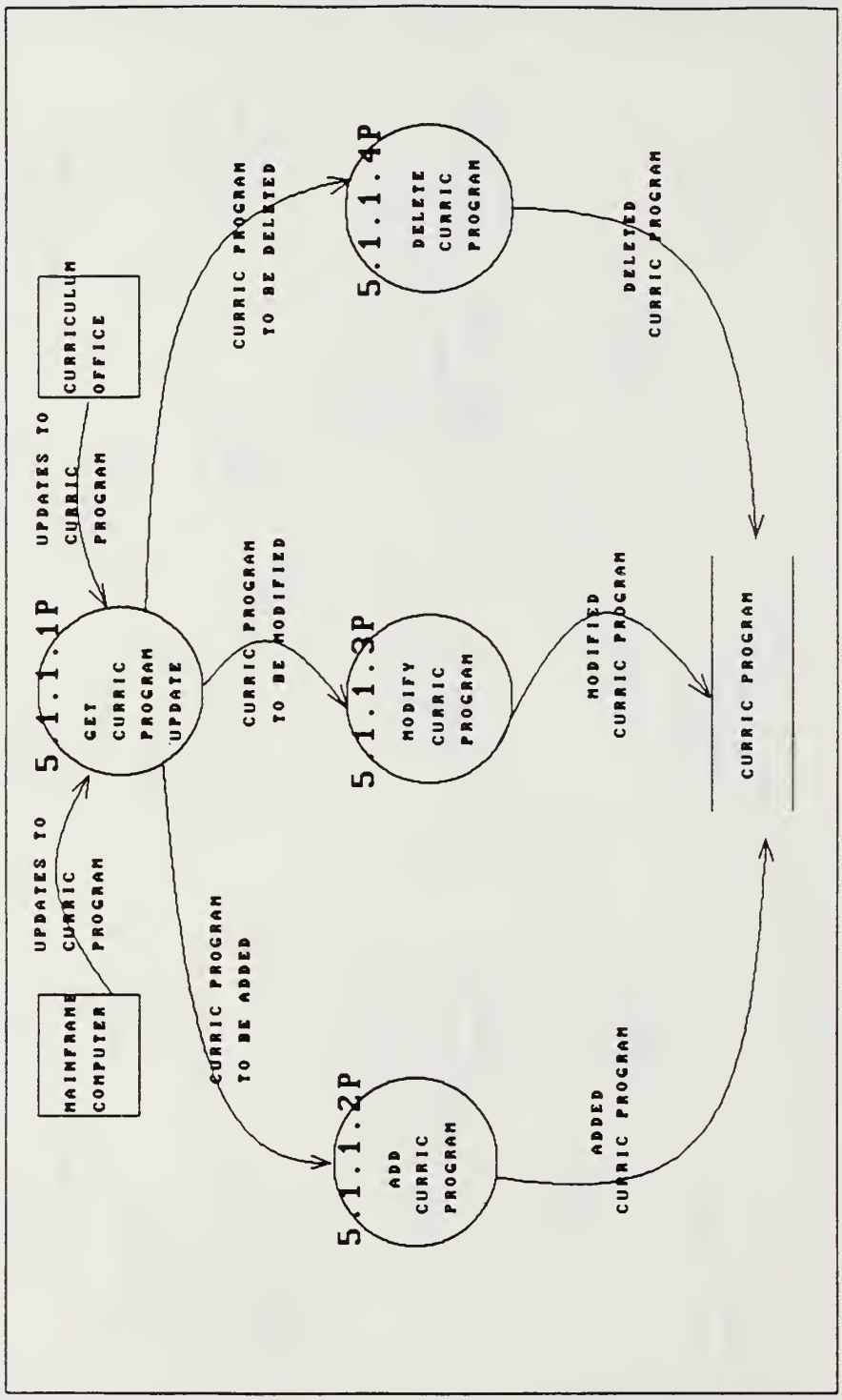


FIG. C-31, Logical Dataflow Diagram for CURRIC PROGRAM Data Store Maintenance.

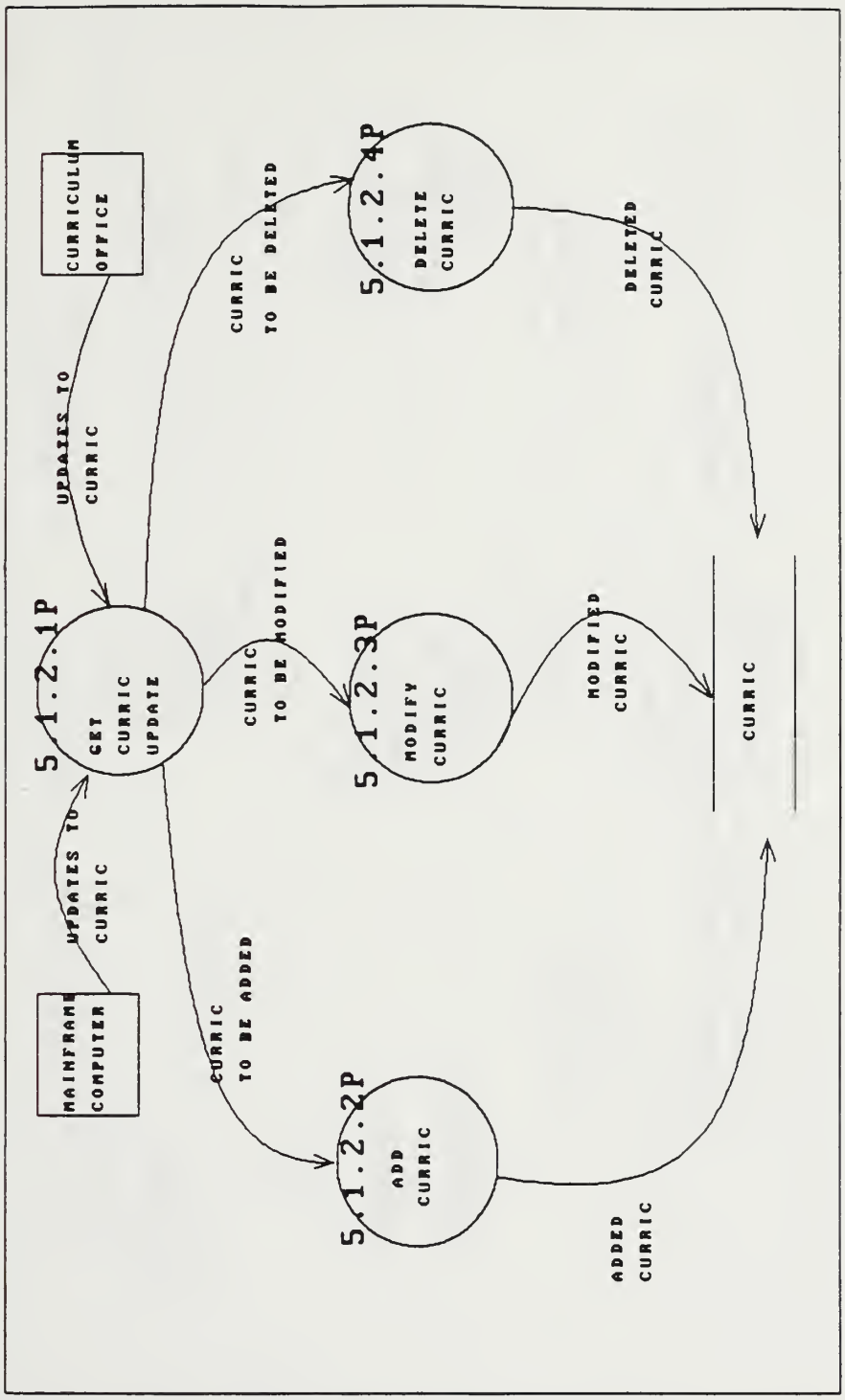


FIG. C-32, Logical Dataflow Diagram for CURRIC Data Store Maintenance.

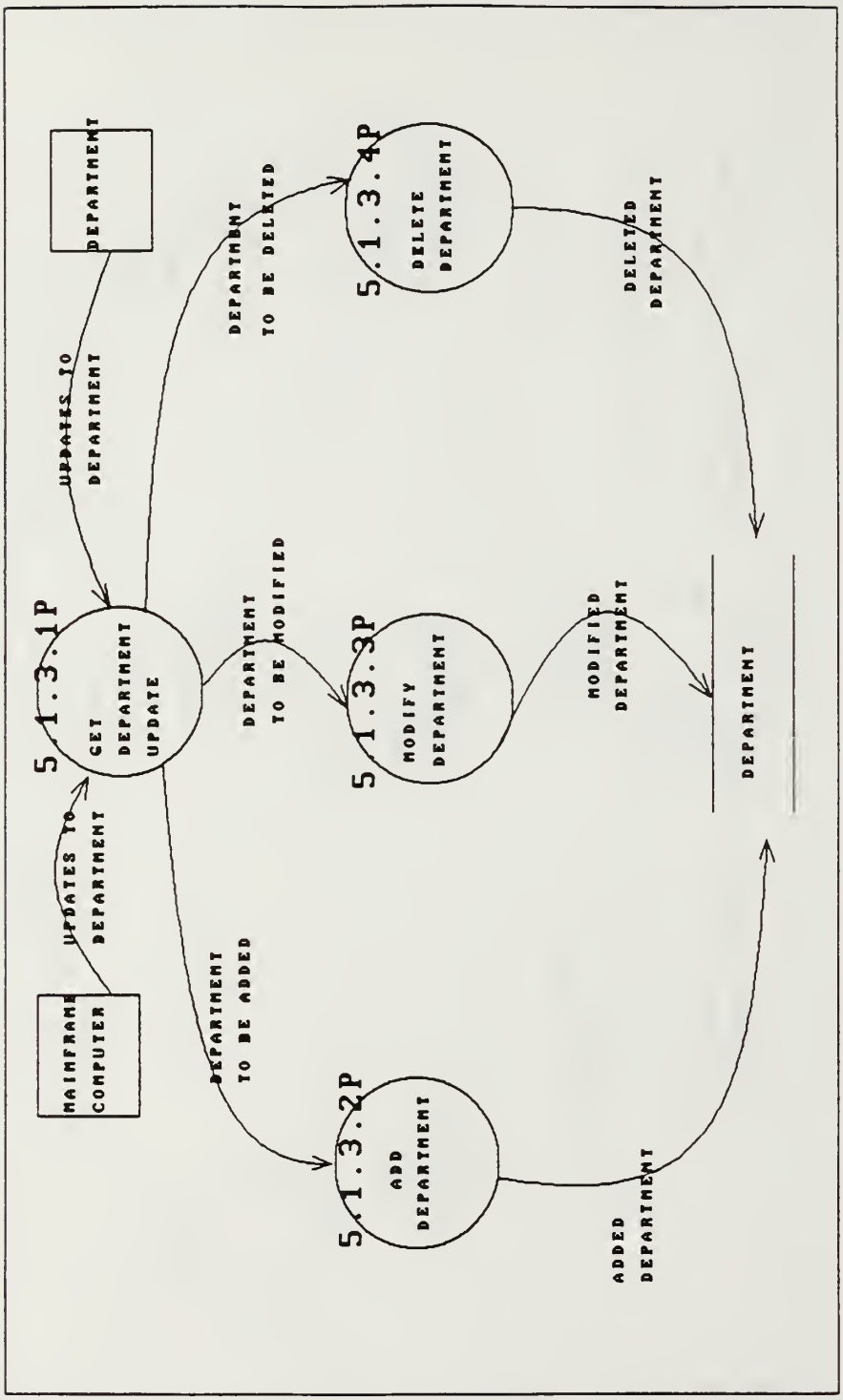


FIG. C-33, Logical Dataflow Diagram for DEPARTMENT Data Store Maintenance.

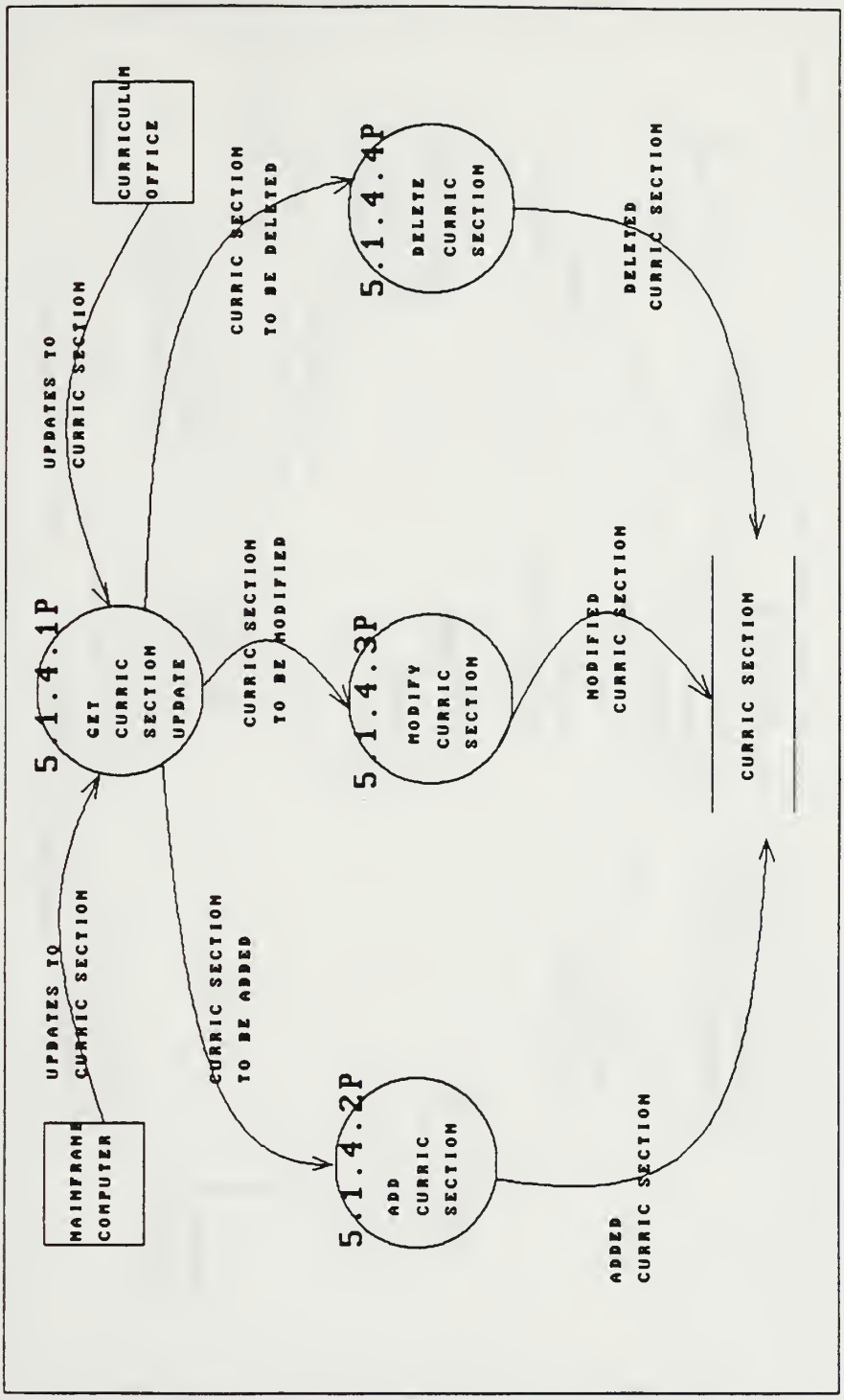


FIG. C-34, Logical Dataflow Diagram for CURRIC SECTION Data Store Maintenance.

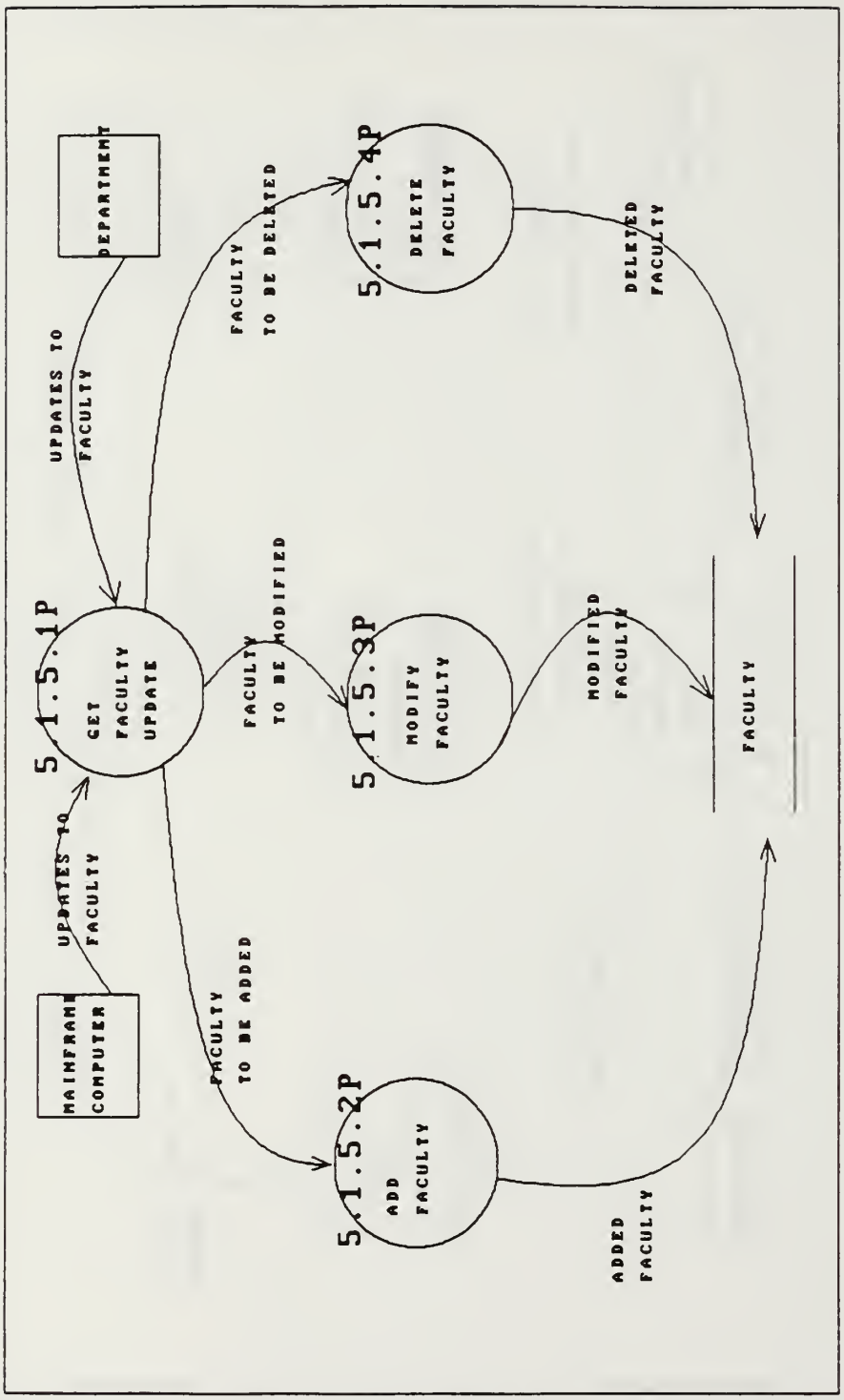


FIG. C-35, Logical Dataflow Diagram for FACULTY Data Store Maintenance.

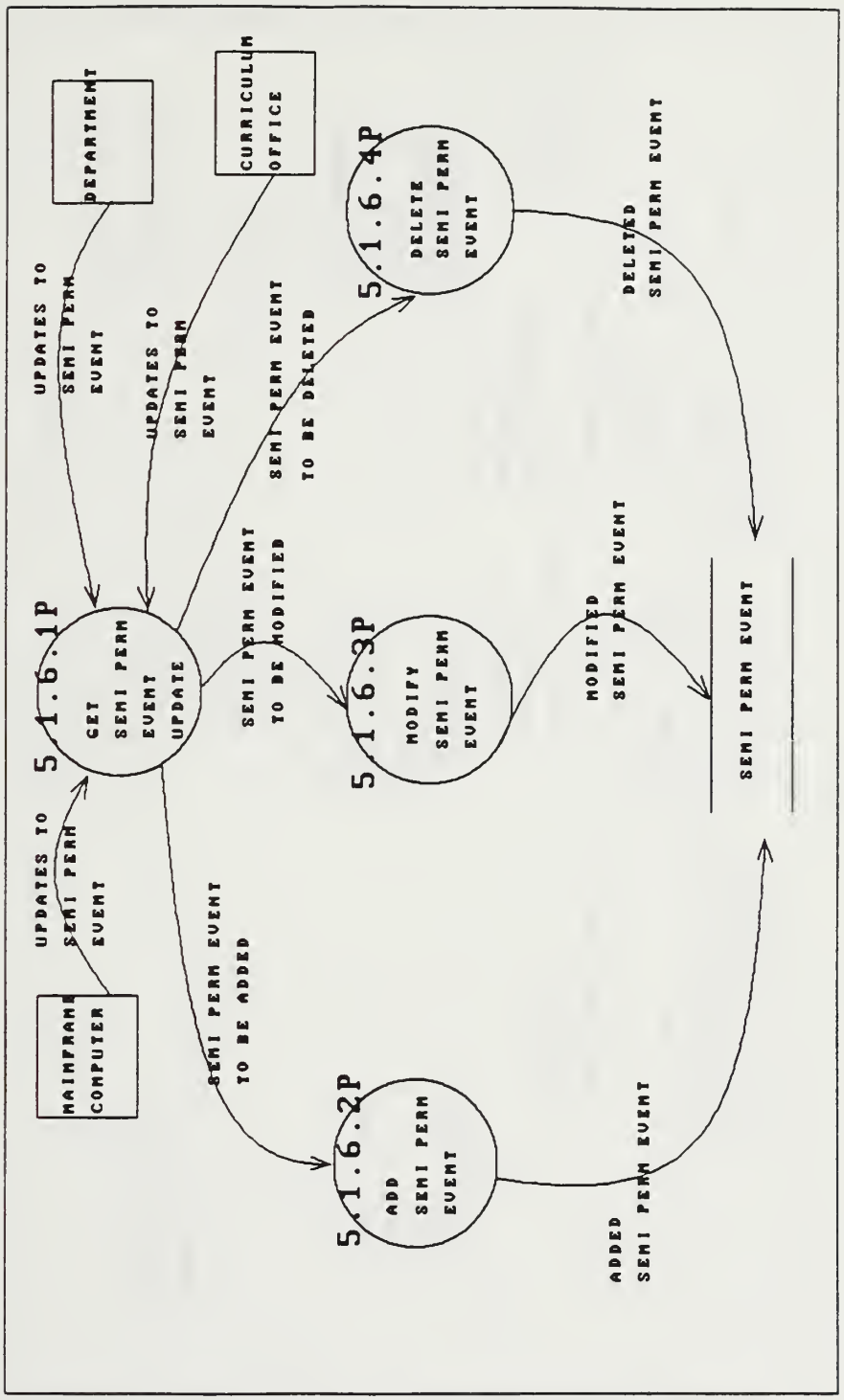


FIG. C-36, Logical Dataflow Diagram for SEMI PERM EVENT Data Store Maintenance.

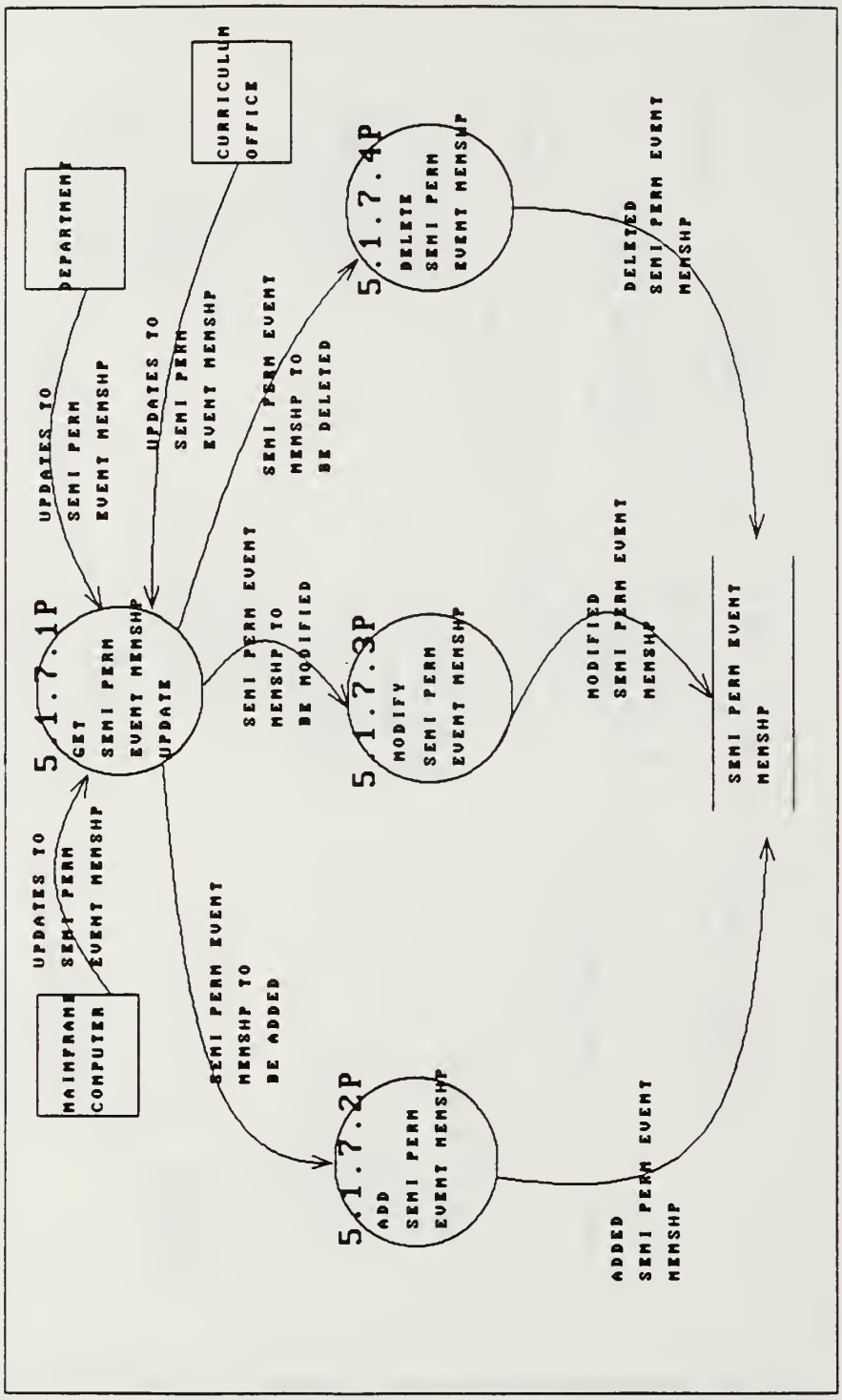

DEPARTMENT
CURRICULUM OFFICE
MAINFRAME COMPUTER
5.1.7.1P
GET SEMI PERM EVENT MEMSHP UPDATE
5.1.7.2P
ADD SEMI PERM EVENT MEMSHP
5.1.7.3P
MODIFY SEMI PERM EVENT MEMSHP
5.1.7.4P
DELETE SEMI PERM EVENT MEMSHP
SEMI PERM EVENT MEMSHP
UPDATES TO SEMI PERM EVENT MEMSHP
UPDATES TO SEMI PERM EVENT MEMSHP
UPDATES TO SEMI PERM EVENT MEMSHP
SEMI PERM EVENT MEMSHP TO BE MODIFIED
SEMI PERM EVENT MEMSHP TO BE DELETED
SEMI PERM EVENT MEMSHP TO BE ADDED
MODIFIED SEMI PERM EVENT MEMSHP
DELETED SEMI PERM EVENT MEMSHP
ADDED SEMI PERM EVENT MEMSHP

FIG. C-37, Logical Dataflow Diagram for SEMI PERM EVENT MEMSHP Data Store Maintenance.

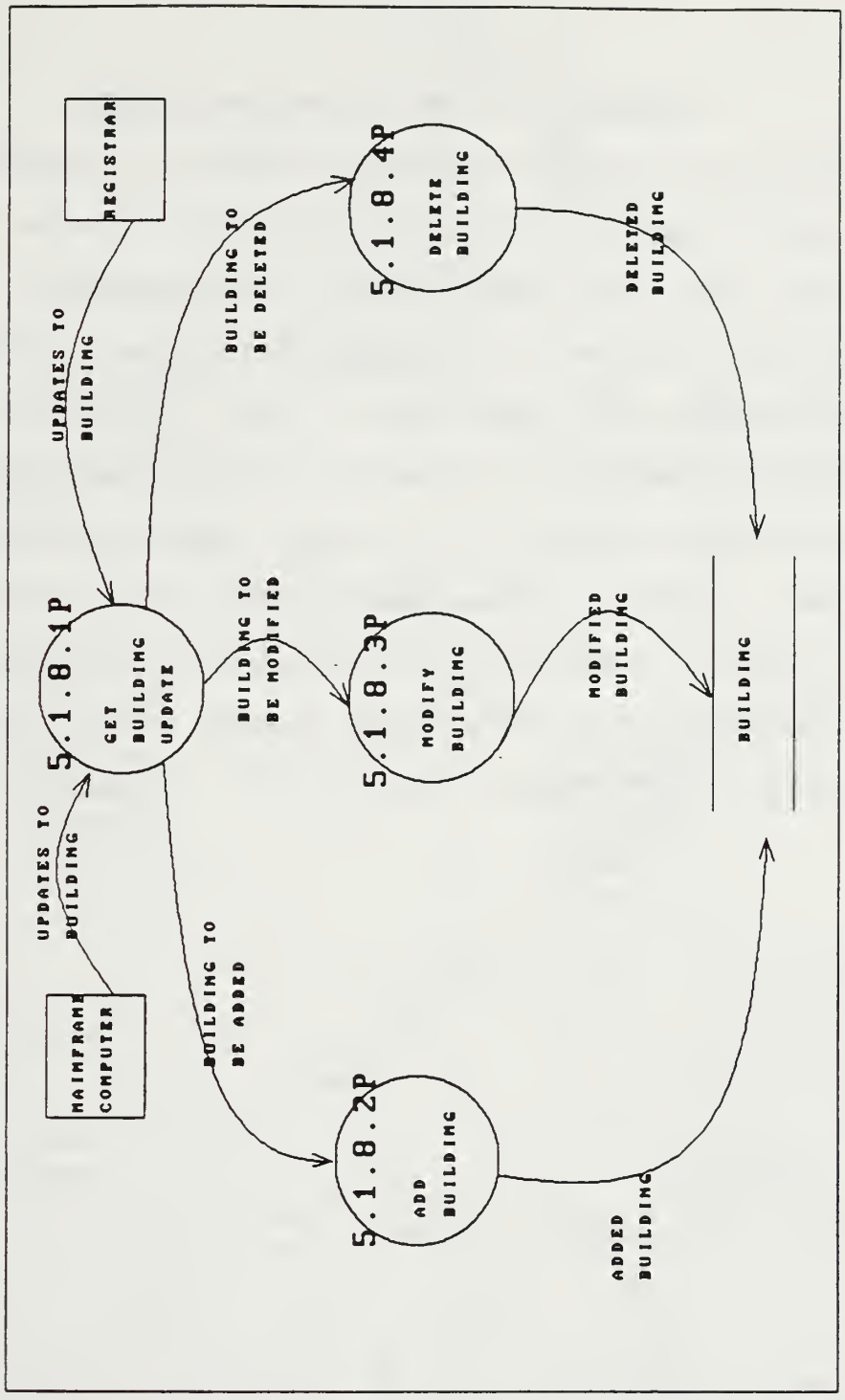


FIG. C-38, Logical Dataflow Diagram for BUILDING Data Store Maintenance.

APPENDIX D: NPS^3 STRUCTURE CHARTS

The following report graphically depicts a computer system structured design, in the form of structure charts, FIG. D-1 thru FIG D-44, for implementing the requirements for the scheduling of courses at the Naval Postgraduate School.

The design of the Final Exam Scheduling, Master Instruction Schedule and Scheduling Data Graph applications are beyond the scope of this thesis and are therefore not addressed, except to show there organization in FIG. D-1 as #'s 5, 6 and 7, respectively. Although not addressed here, the advantages of structured Design allow for these applications to be easily added to NPS^3 in the future.

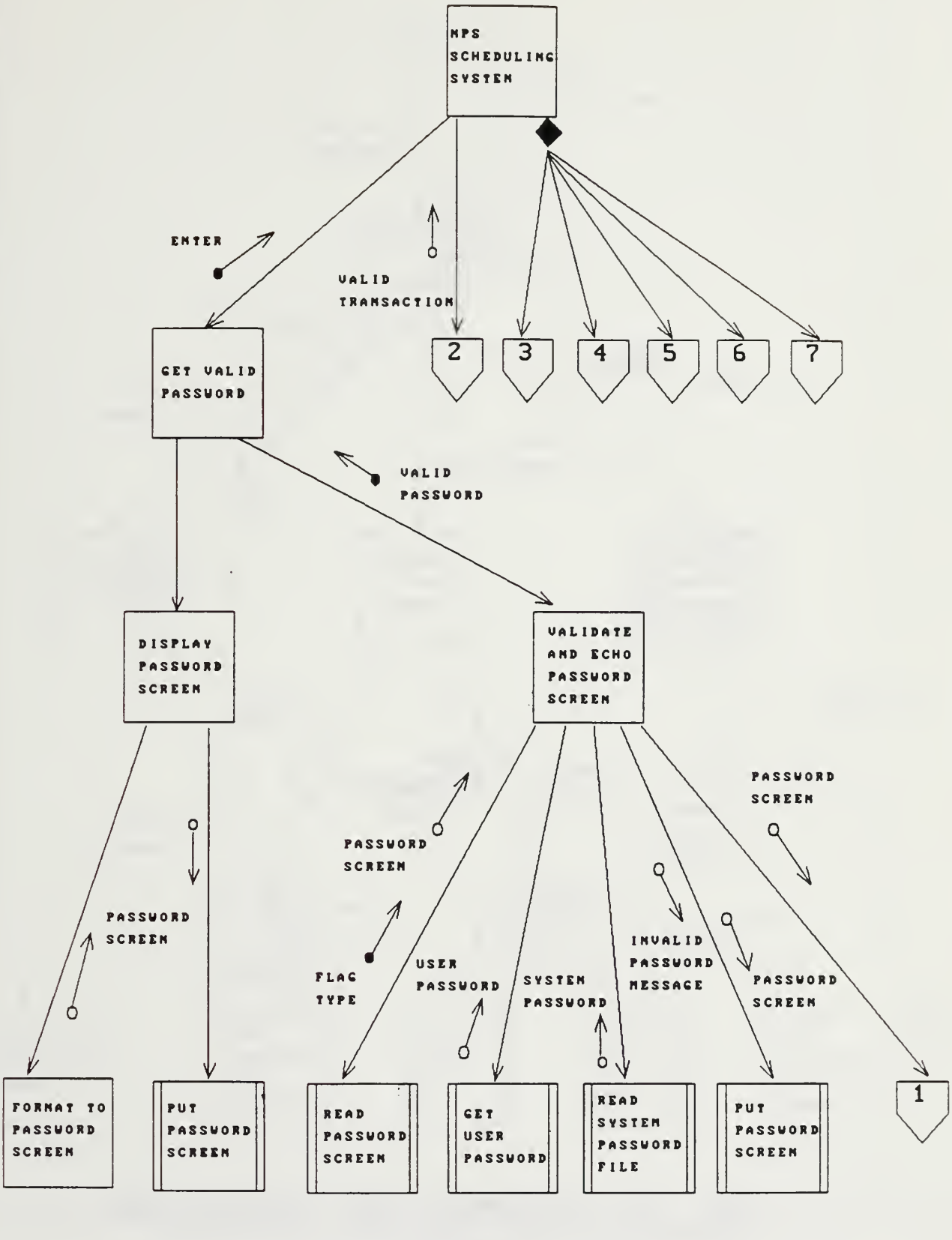


Fig. D-1, NPS SCHEDULING SYSTEM.

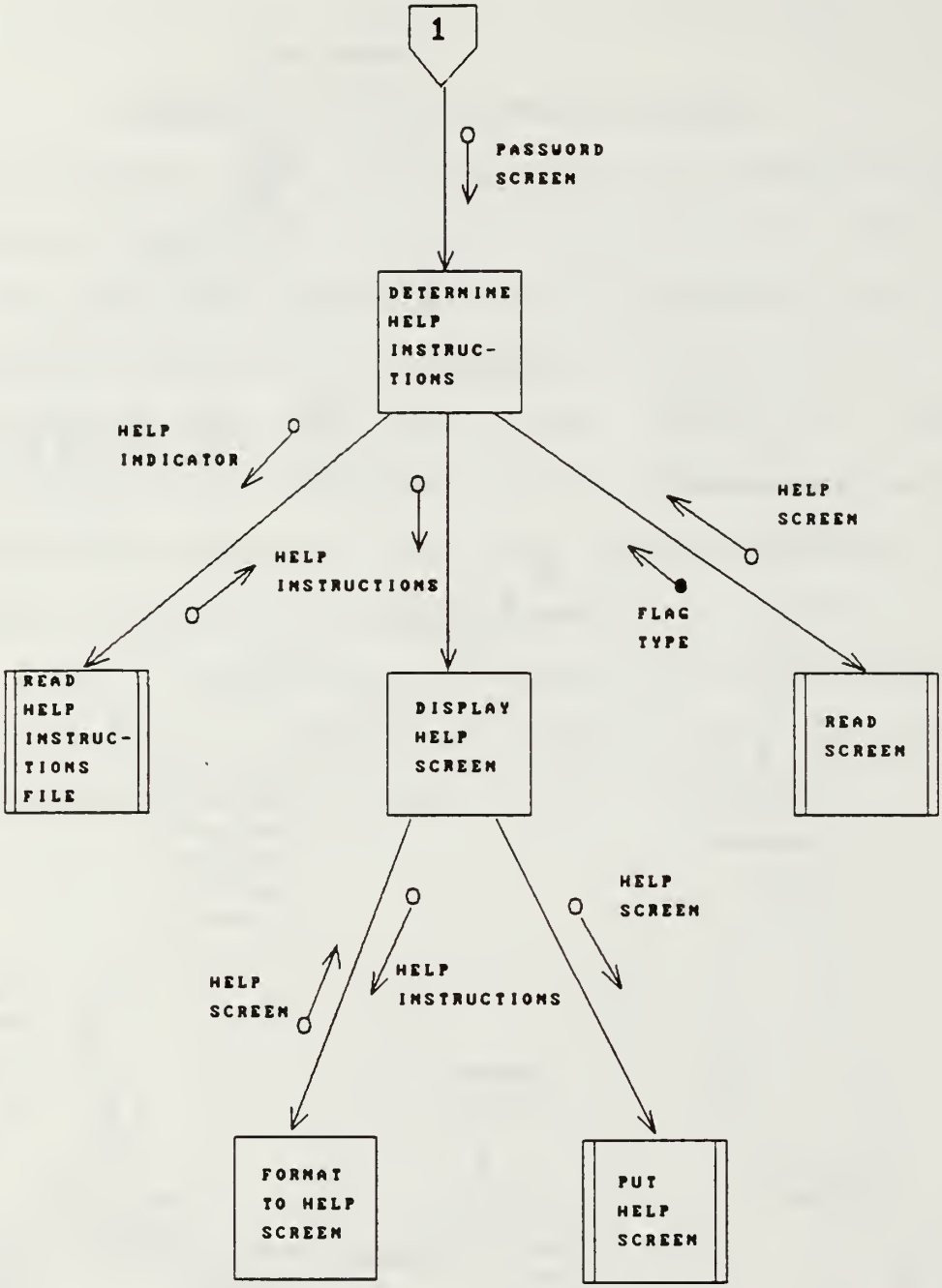


Fig. D-2, DETERMINE HELP INSTRUCTIONS for Password Screen.

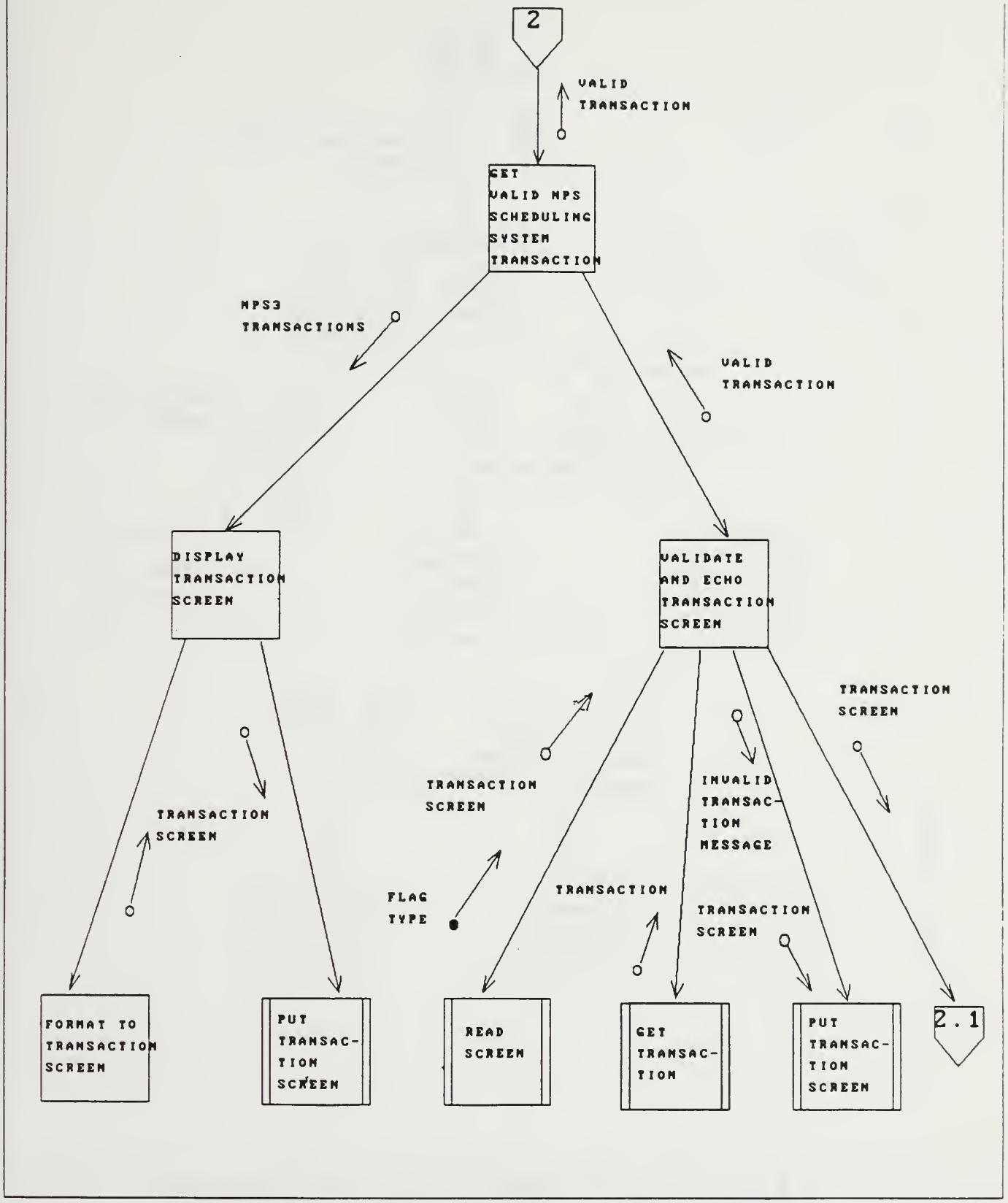


Fig. D-3, GET VALID NPS SCHEDULING SYSTEM TRANSACTION .

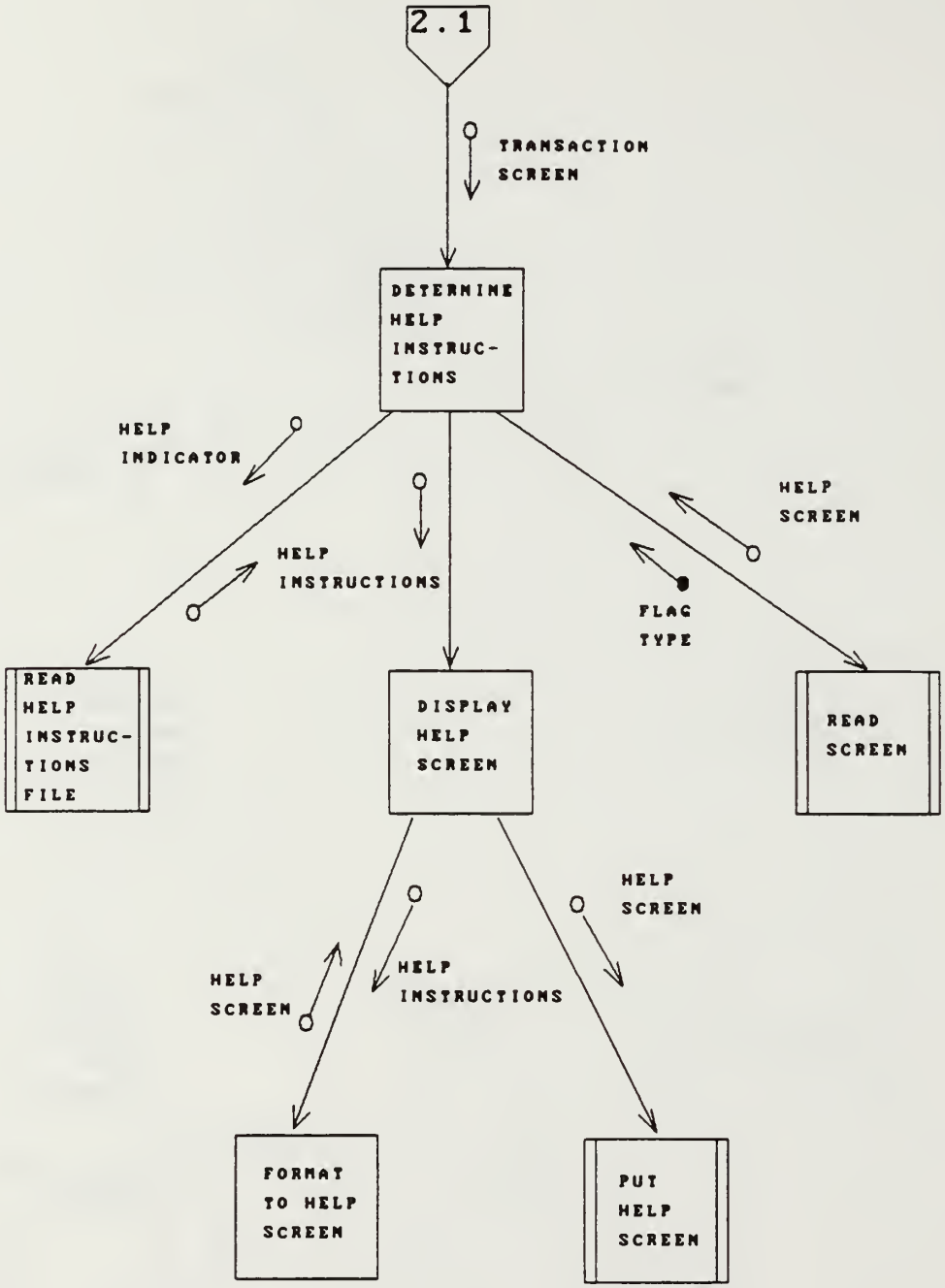


Fig. D-4, DETERMINE HELP INSTRUCTIONS for Transaction Screen.

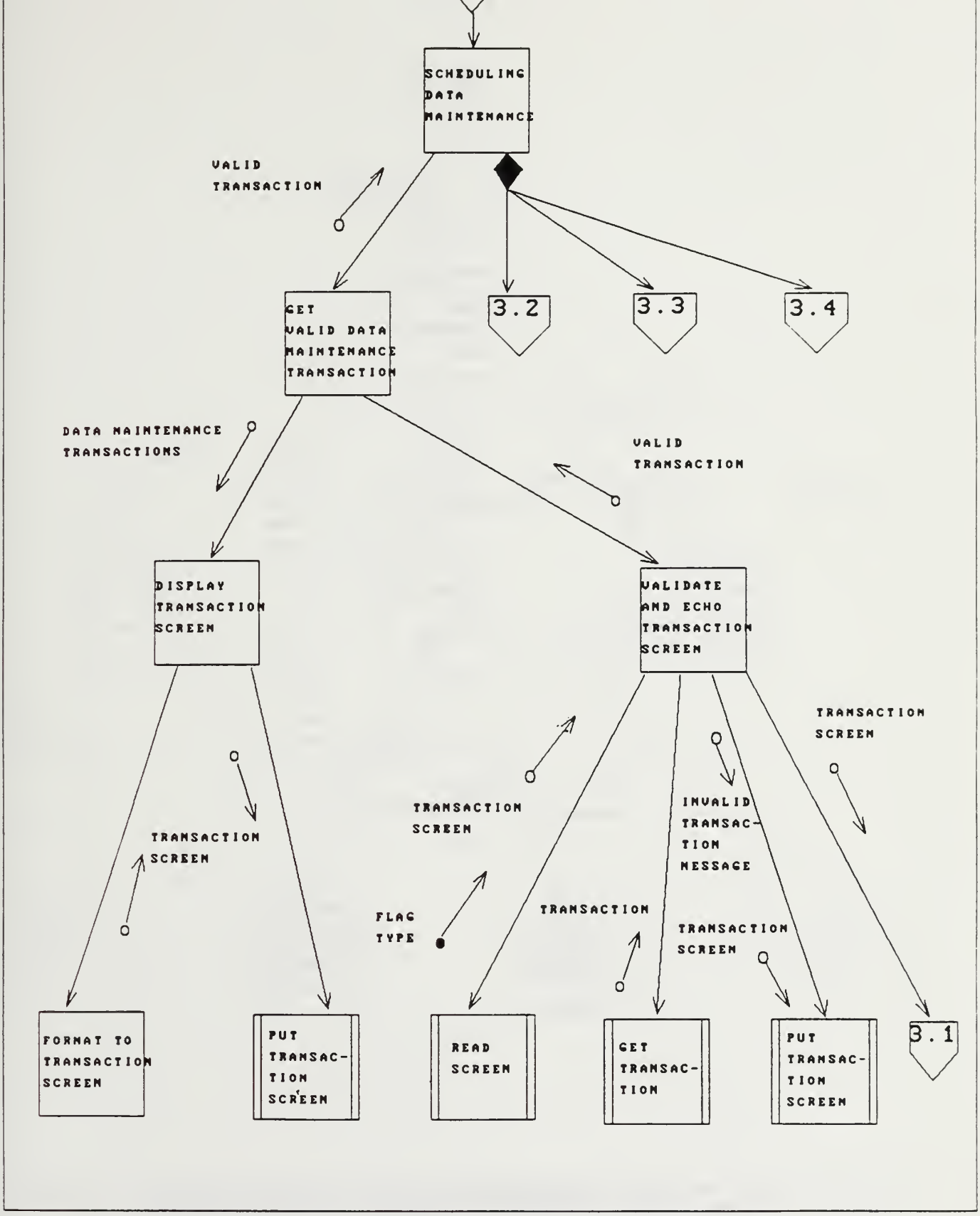


Fig. D-5, SCHEDULING DATA MAINTENANCE.

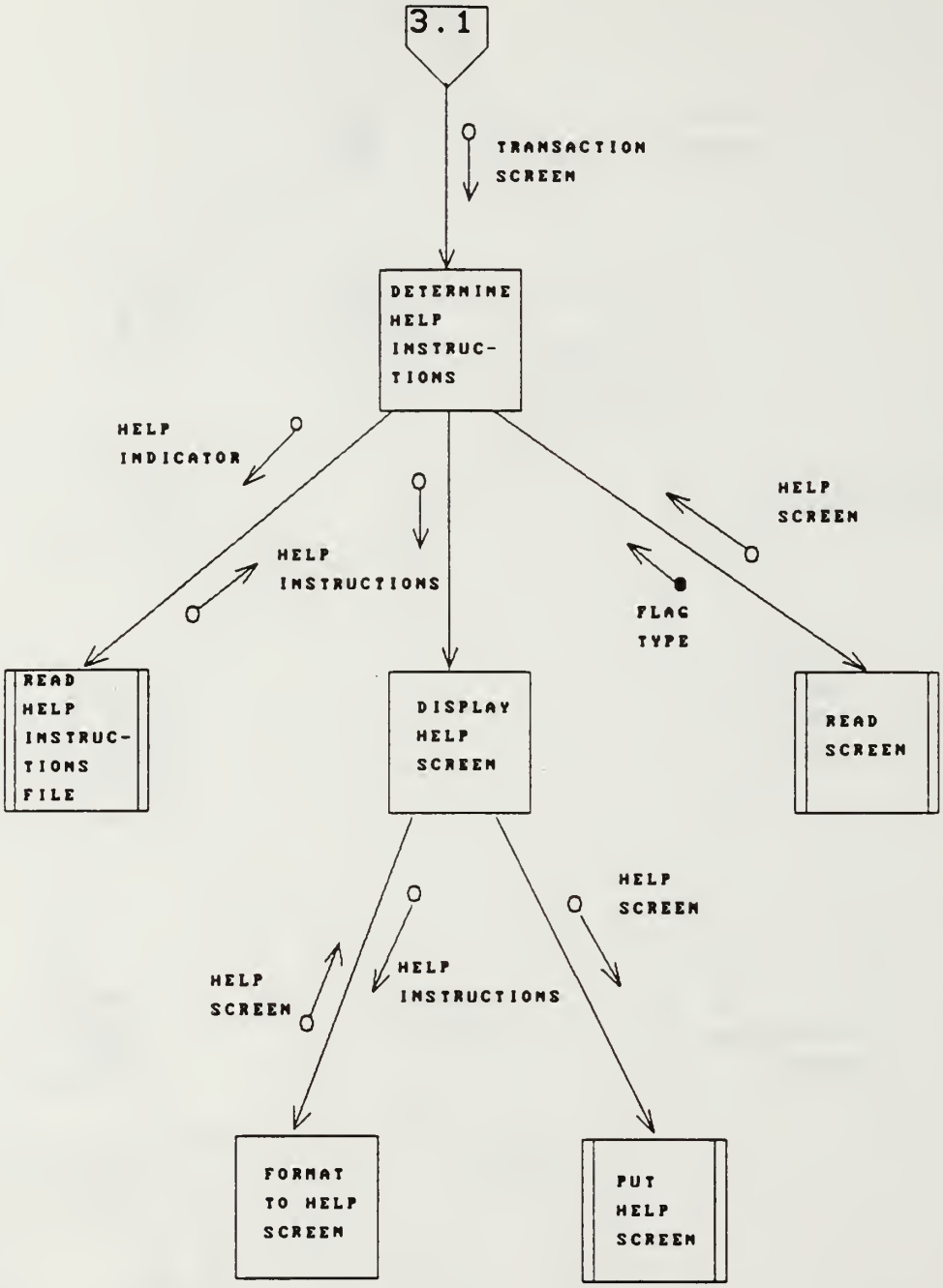


Fig. D-6, DETERMINE HELP INSTRUCTIONS for Transaction Screen.

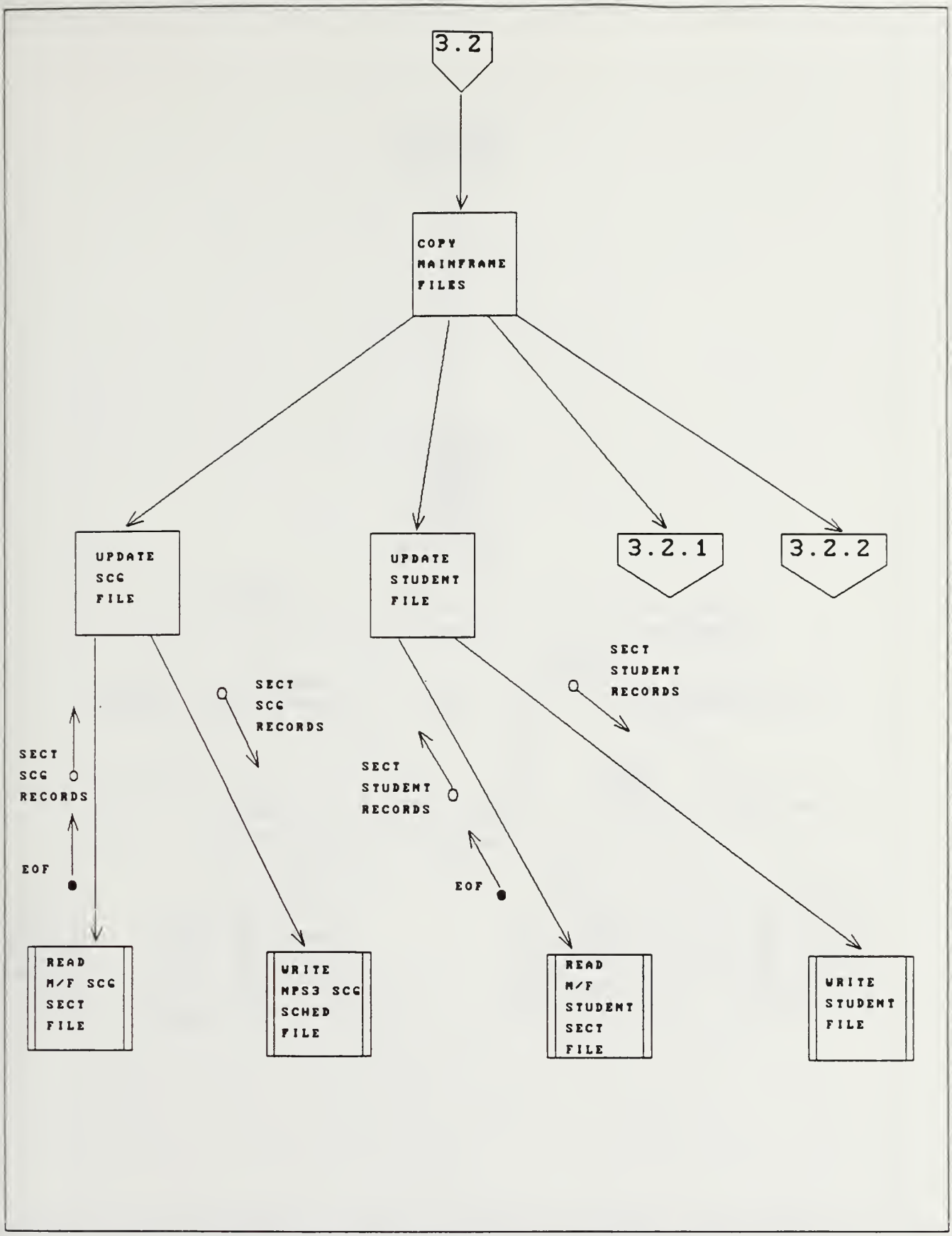


Fig. D-7, COPY MAINFRAME FILES.

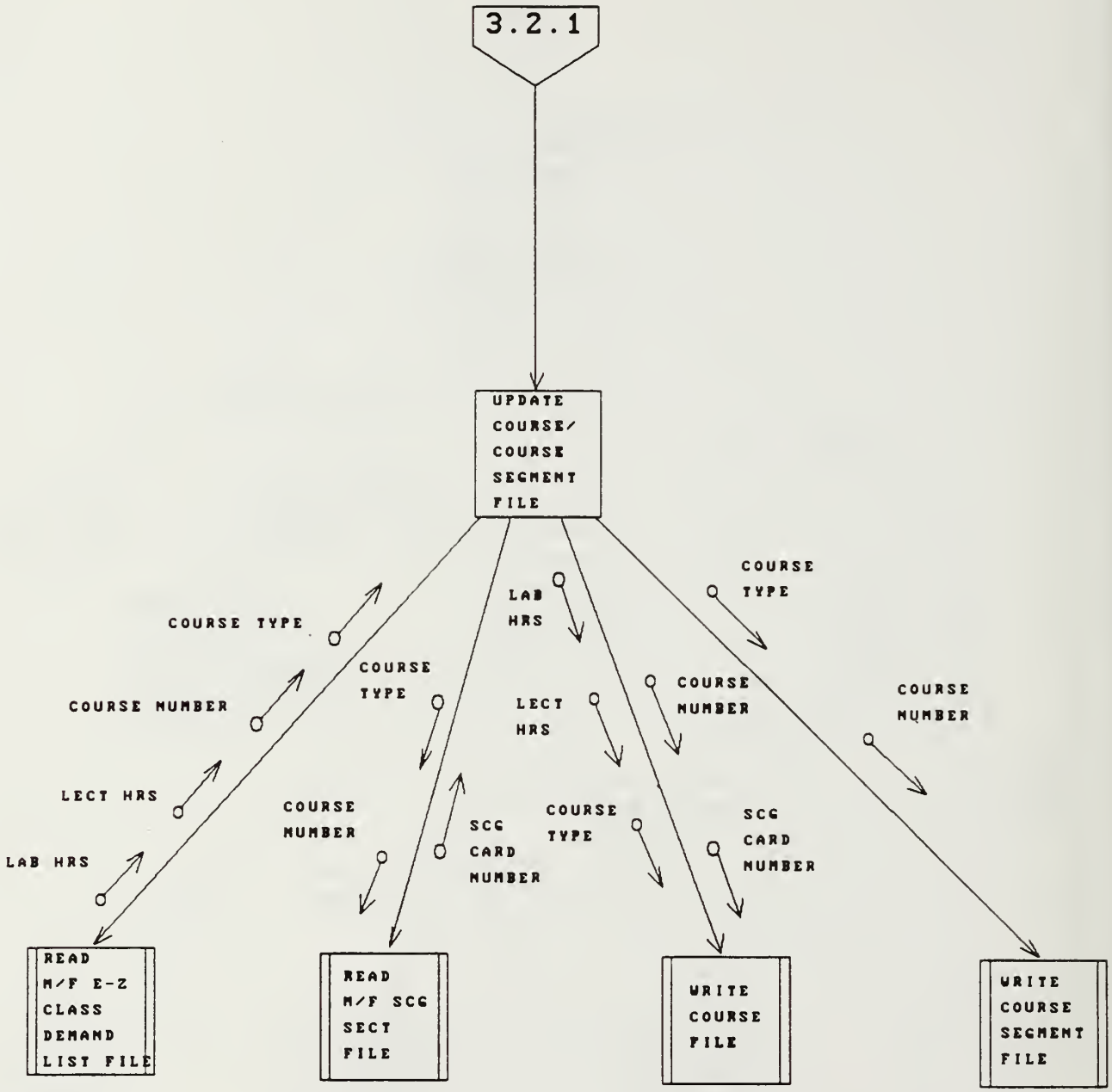


Fig. D-8, UPDATE COURSE/COURSE SEGMENT FILES.

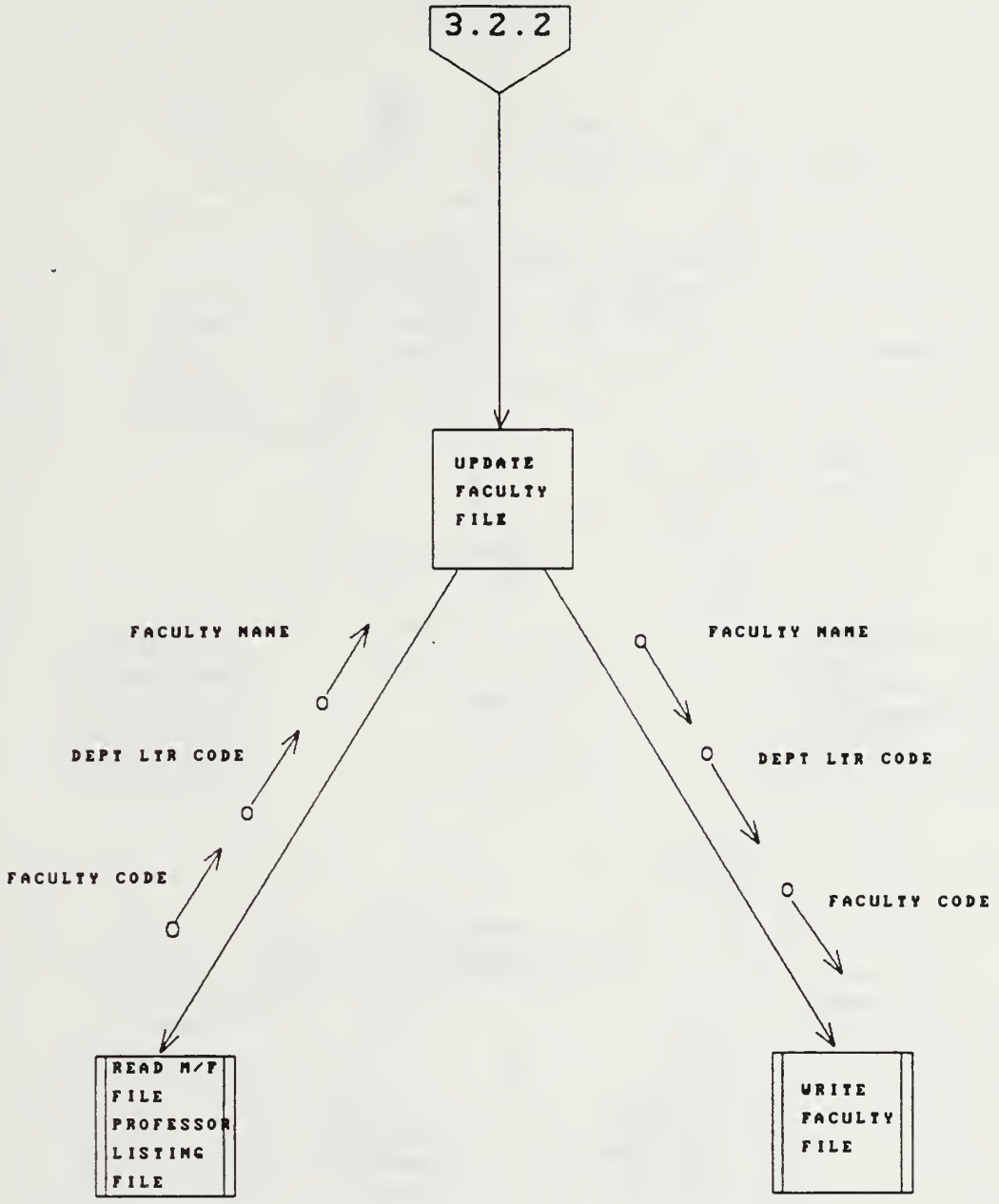


Fig. D-9, UPDATE FACULTY FILE.

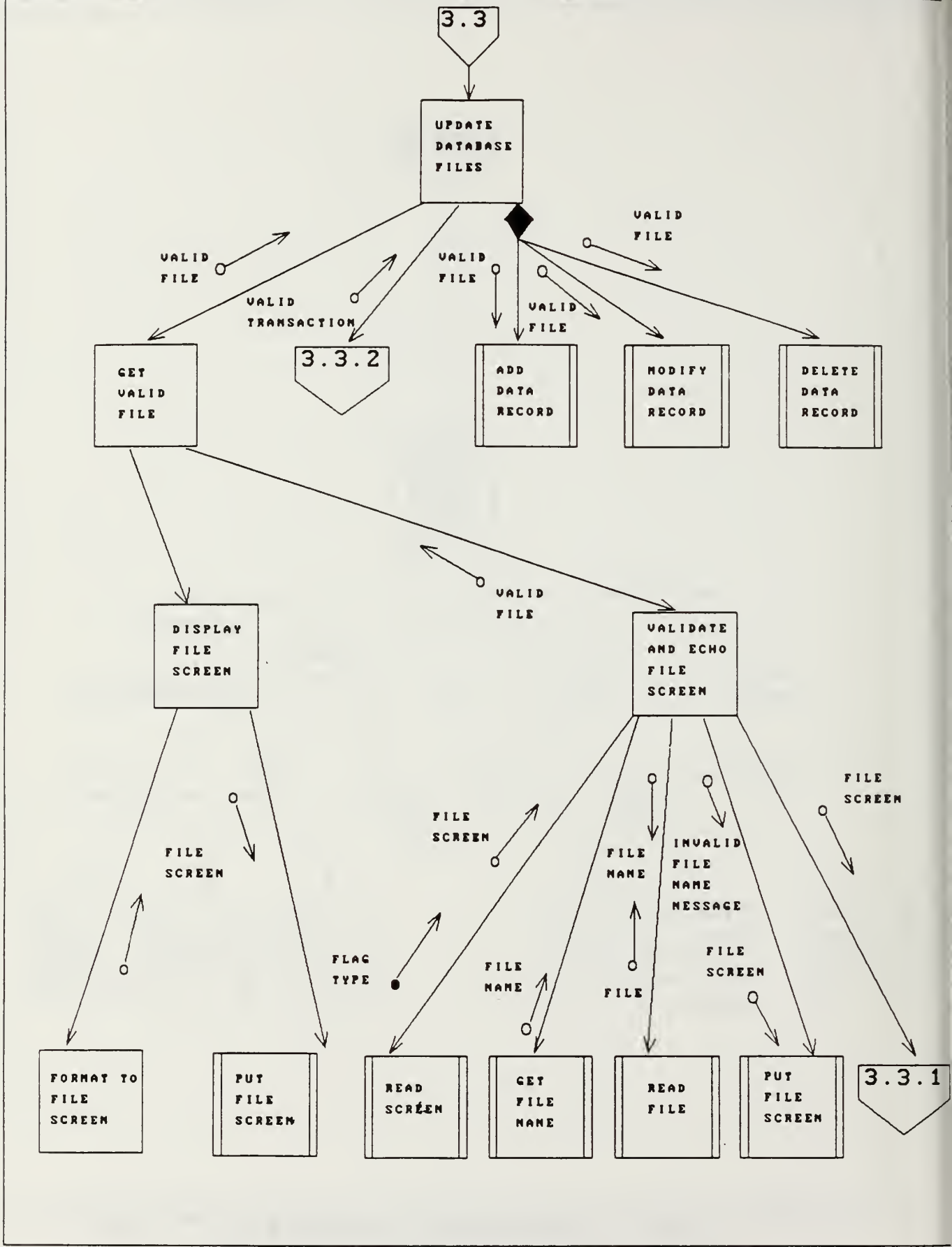


Fig. D-10, UPDATE DATABASE FILES.

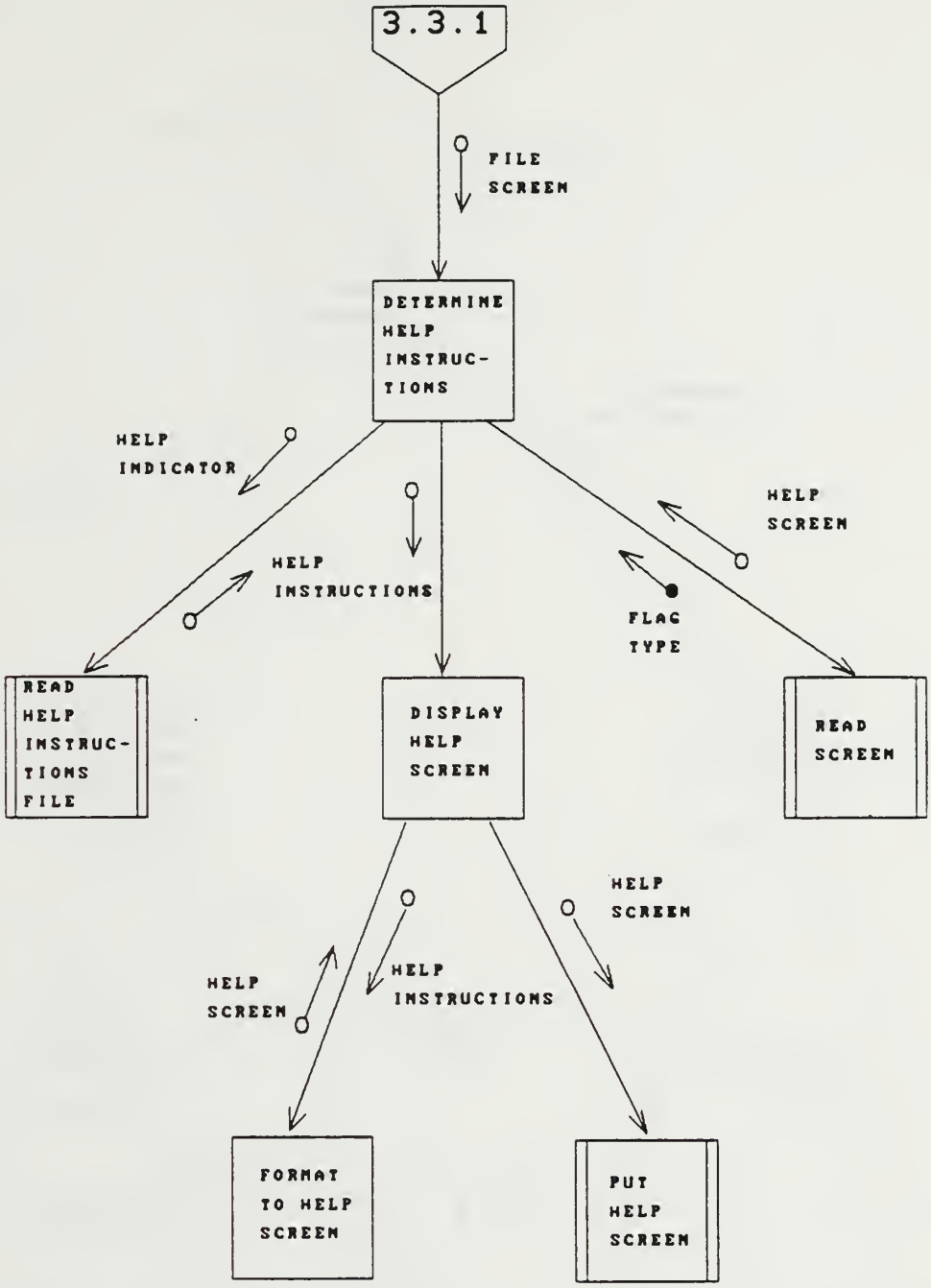


Fig. D-11, DETERMINE HELP INSTRUCTIONS for File Screen.

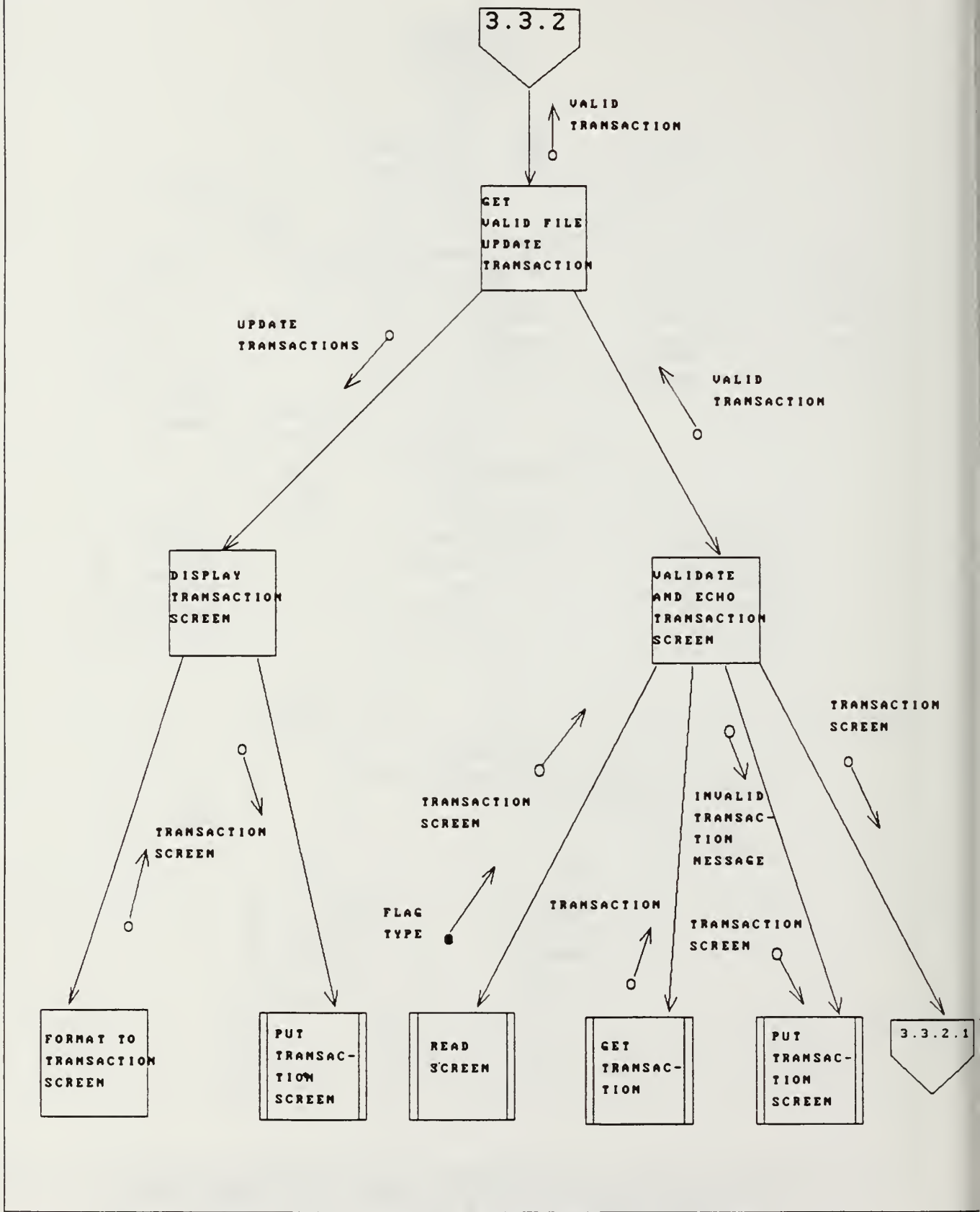


Fig. D-12, GET VALID FILE UPDATE TRANSACTION.

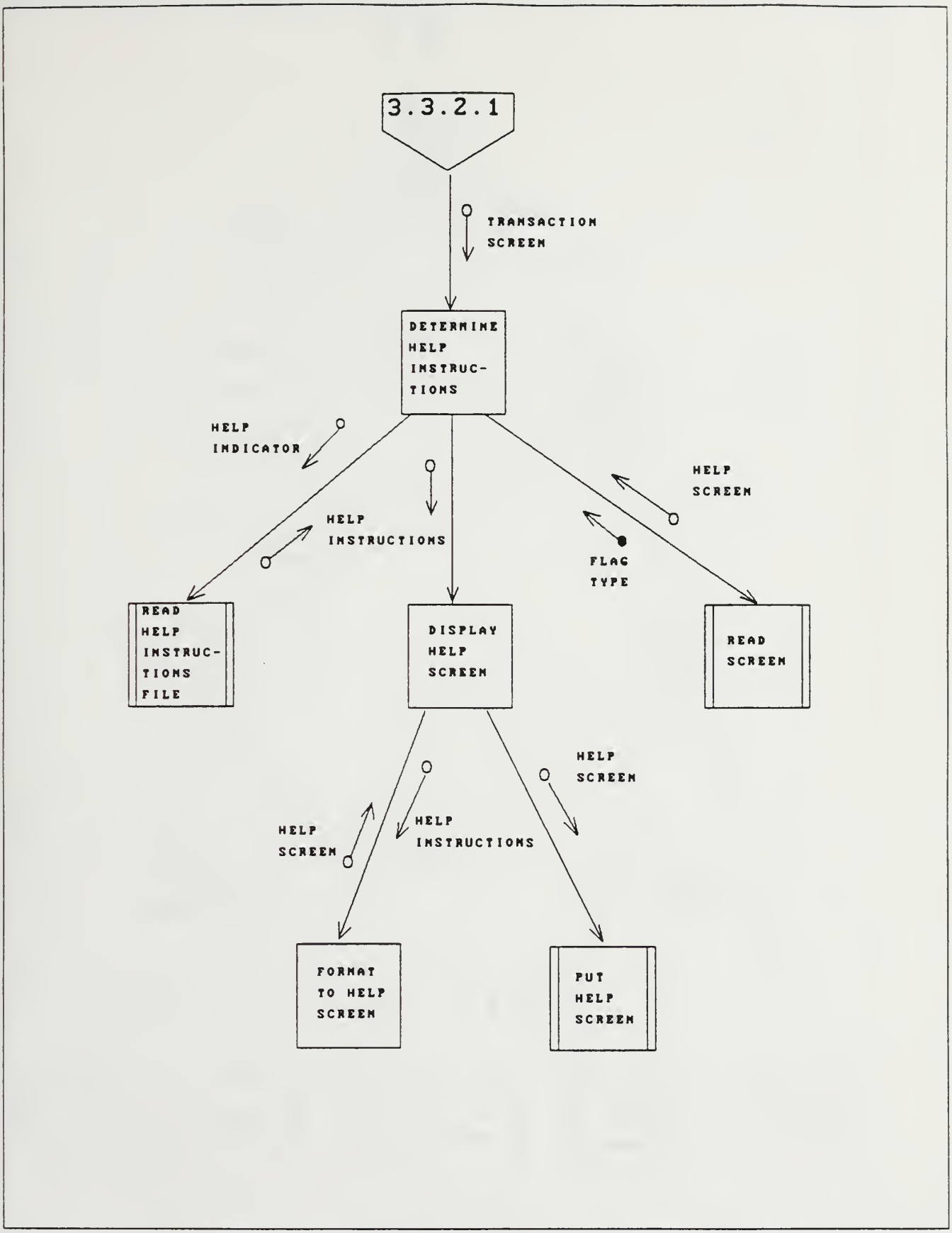


Fig. D-13, DETERMINE HELP INSTRUCTIONS for Transaction Screen.

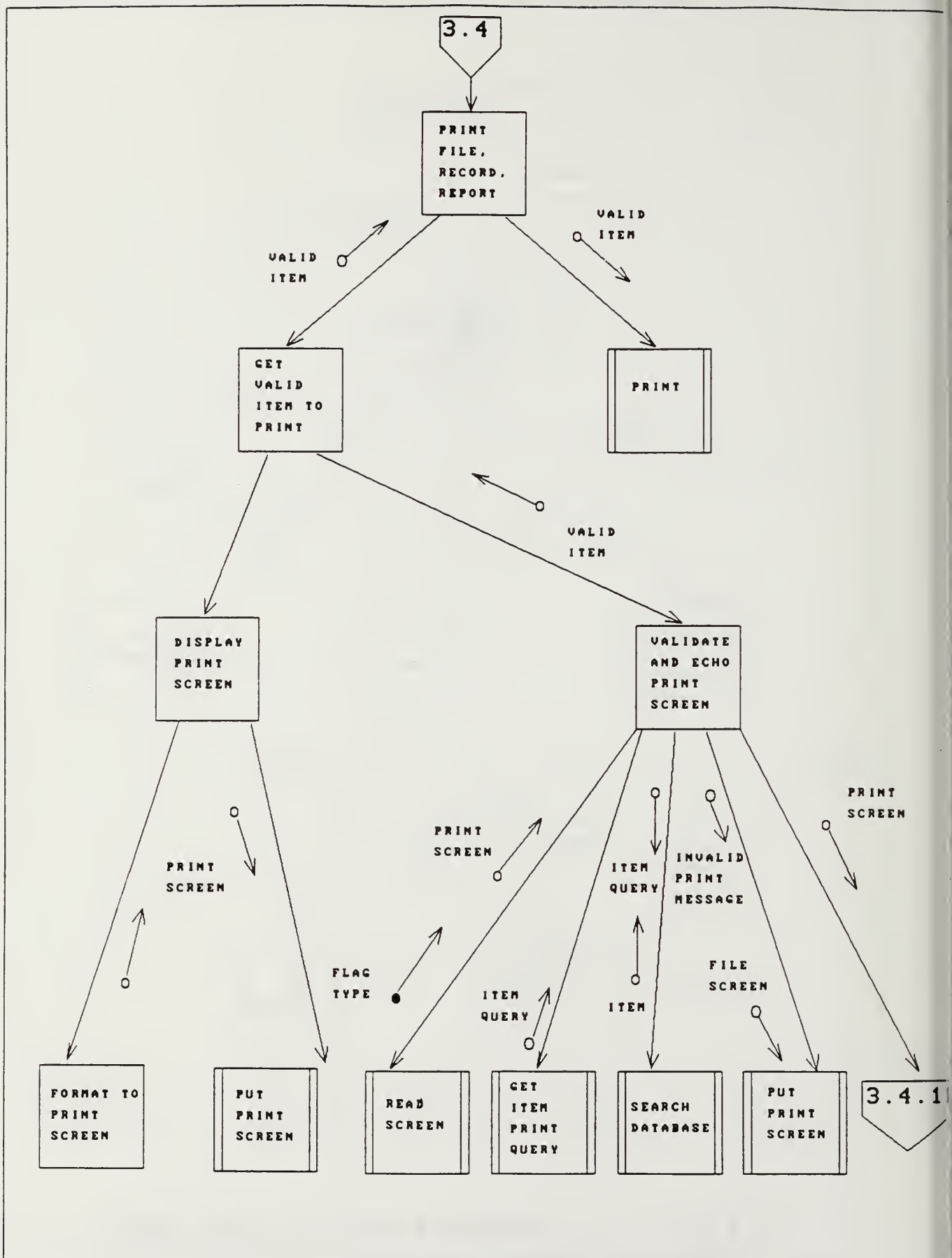


Fig. D-14, PRINT FILE, RECORD, REPORT.

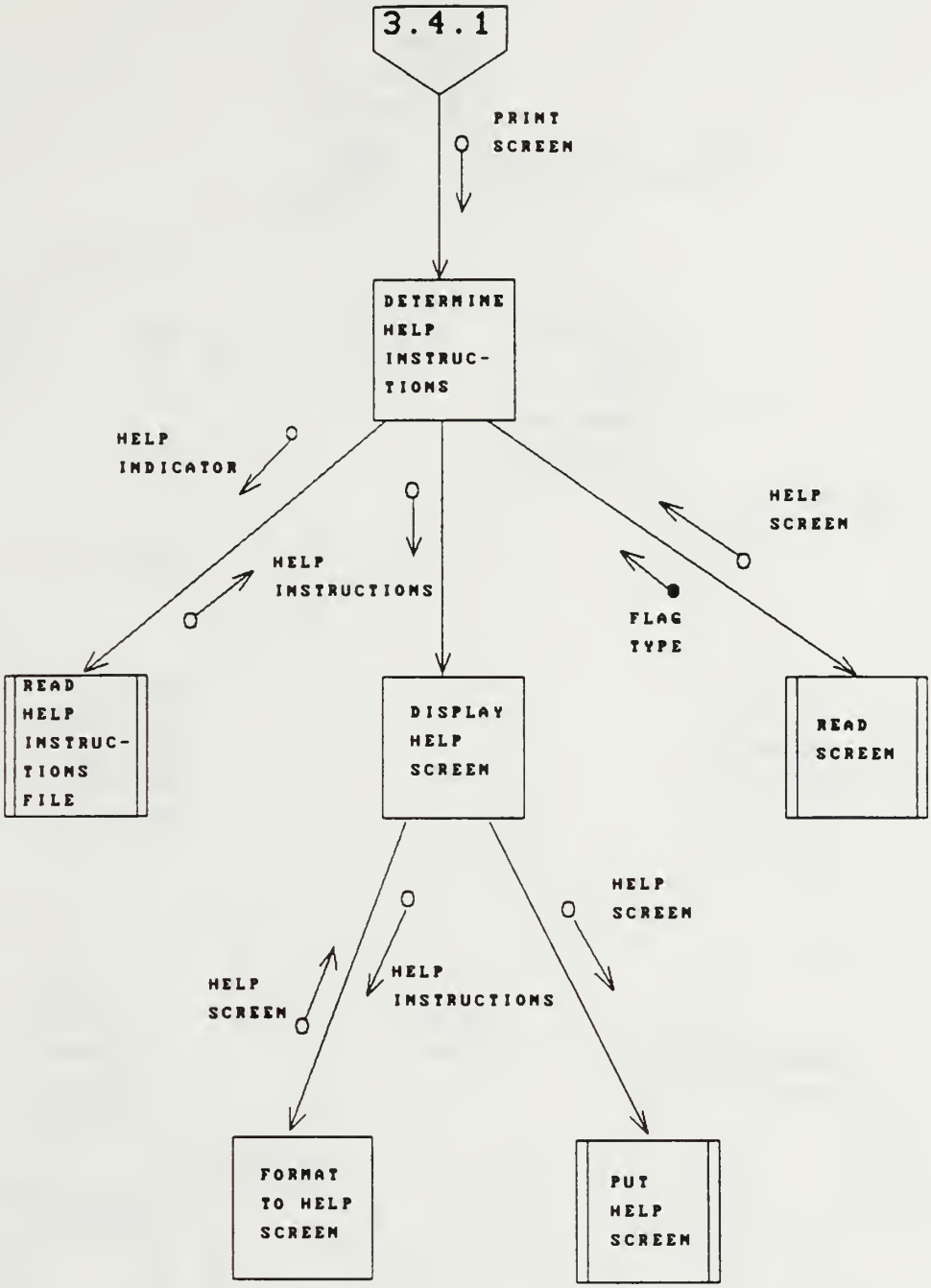


Fig. D-15, DETERMINE HELP INSTRUCTIONS for Print Screen.

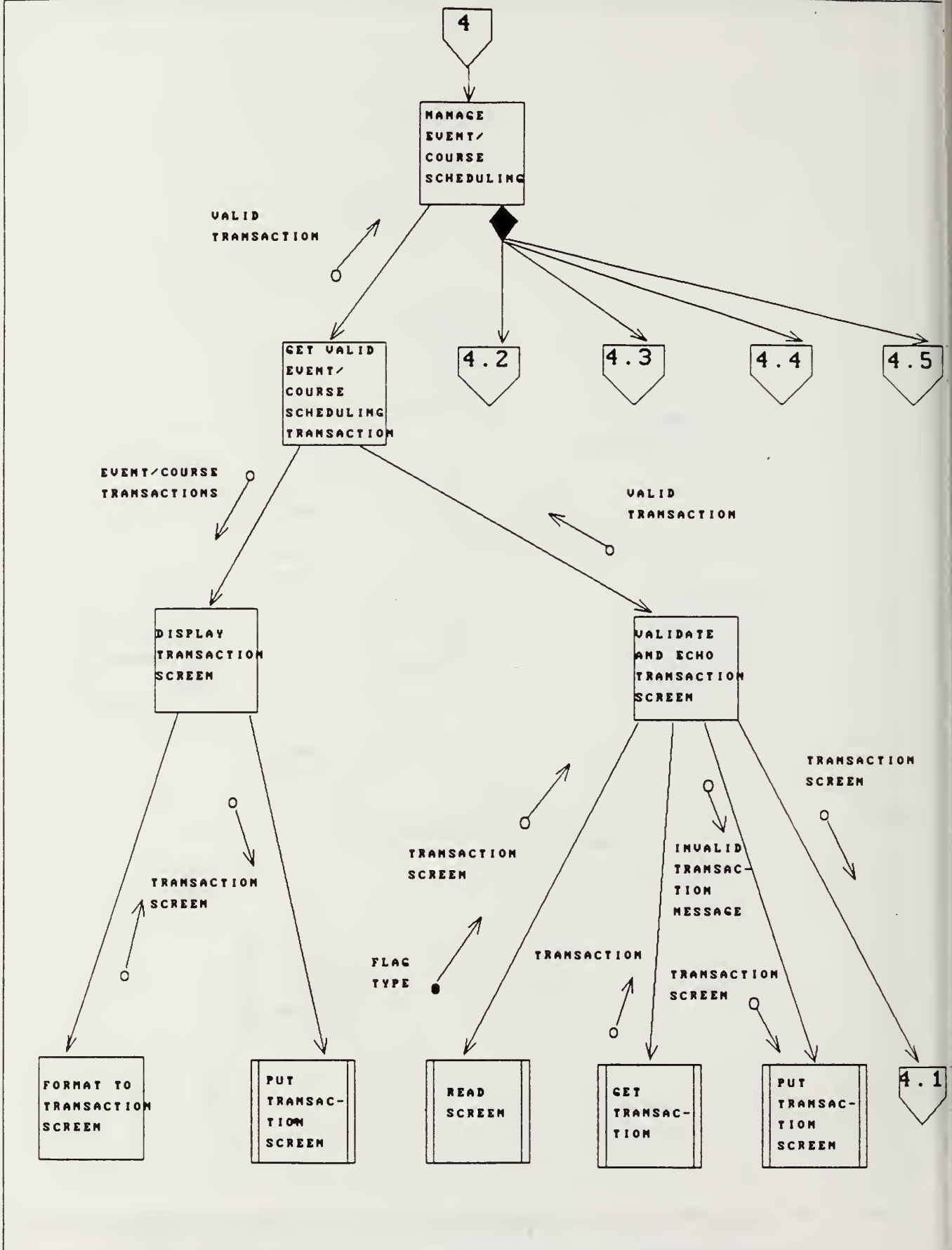


Fig. D-16, MANAGE EVENT/COURSE SCHEDULING.

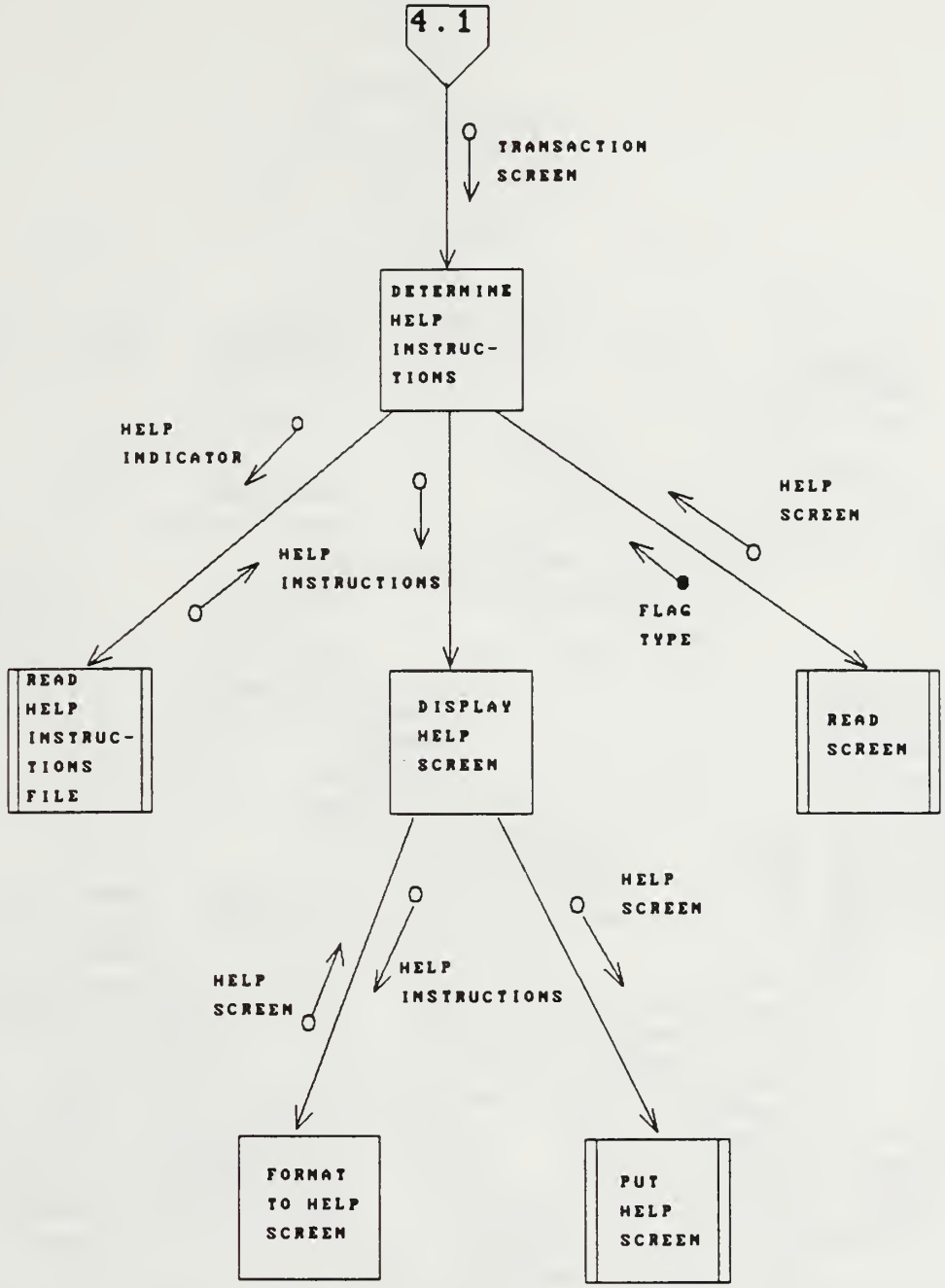


Fig. D-17, DETERMINE HELP INSTRUCTIONS for Event/Course Transaction Screen.

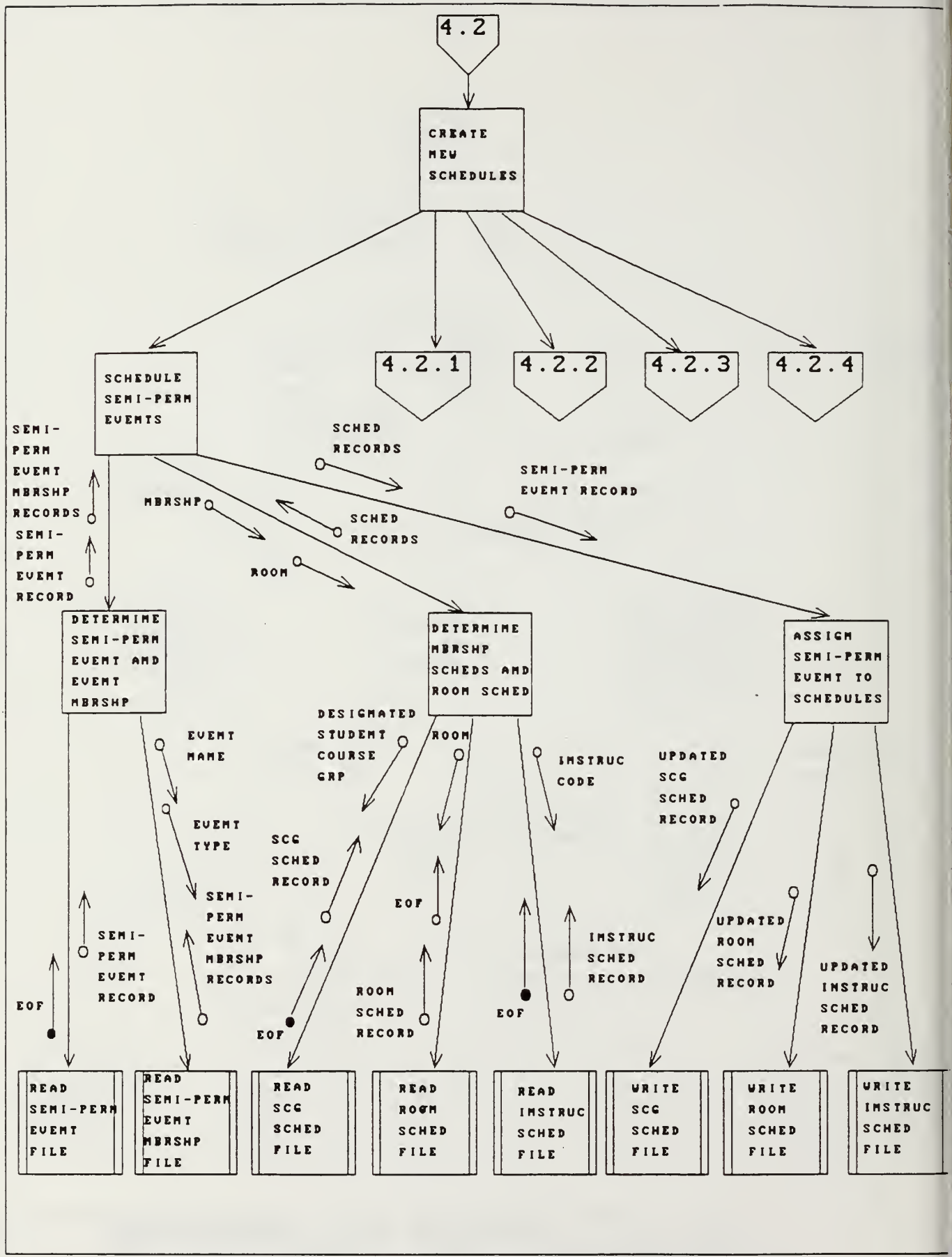


Fig. D-18, CREATE NEW SCHEDULES.

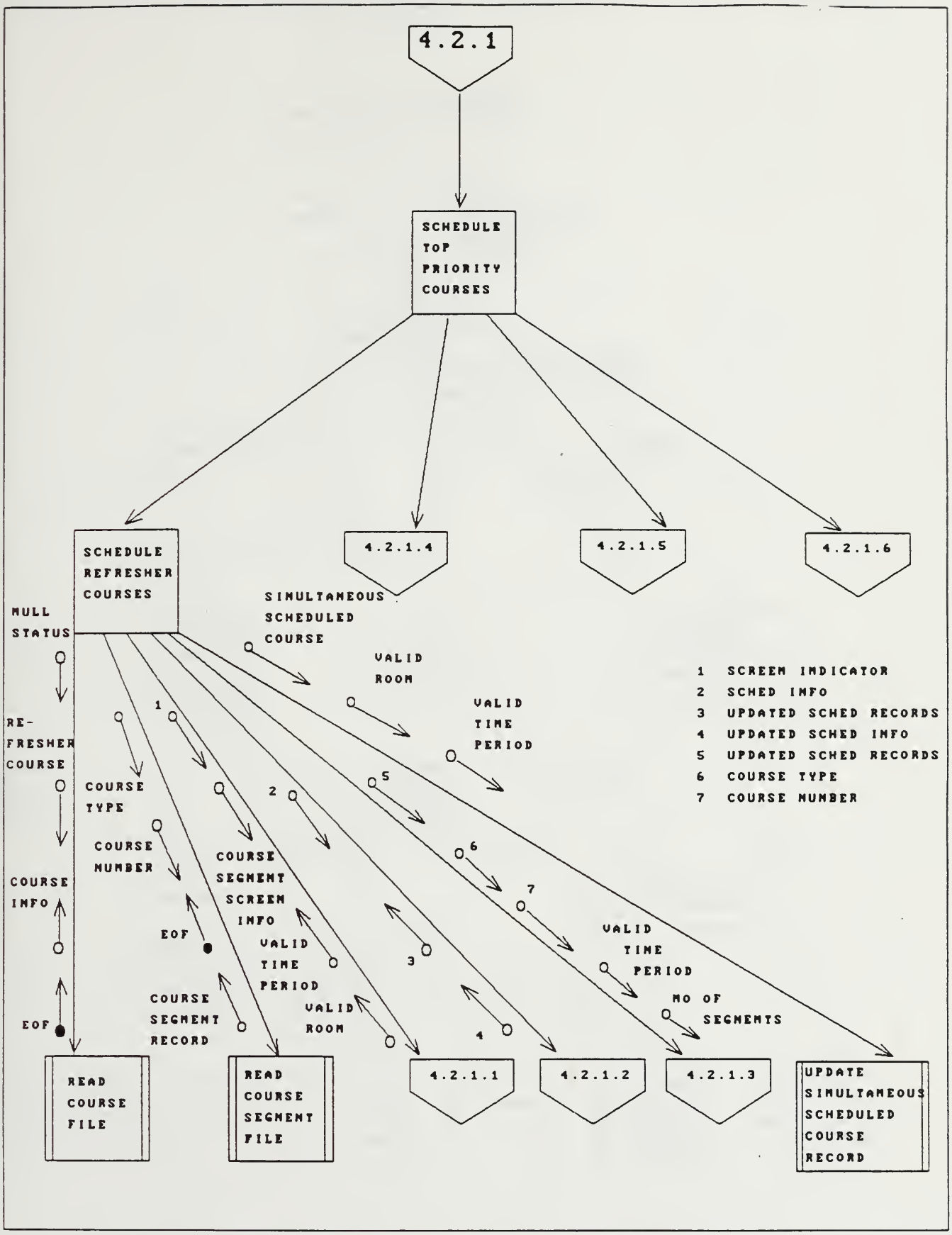


Fig. D-19, SCHEDULE TOP PRIORITY COURSES.

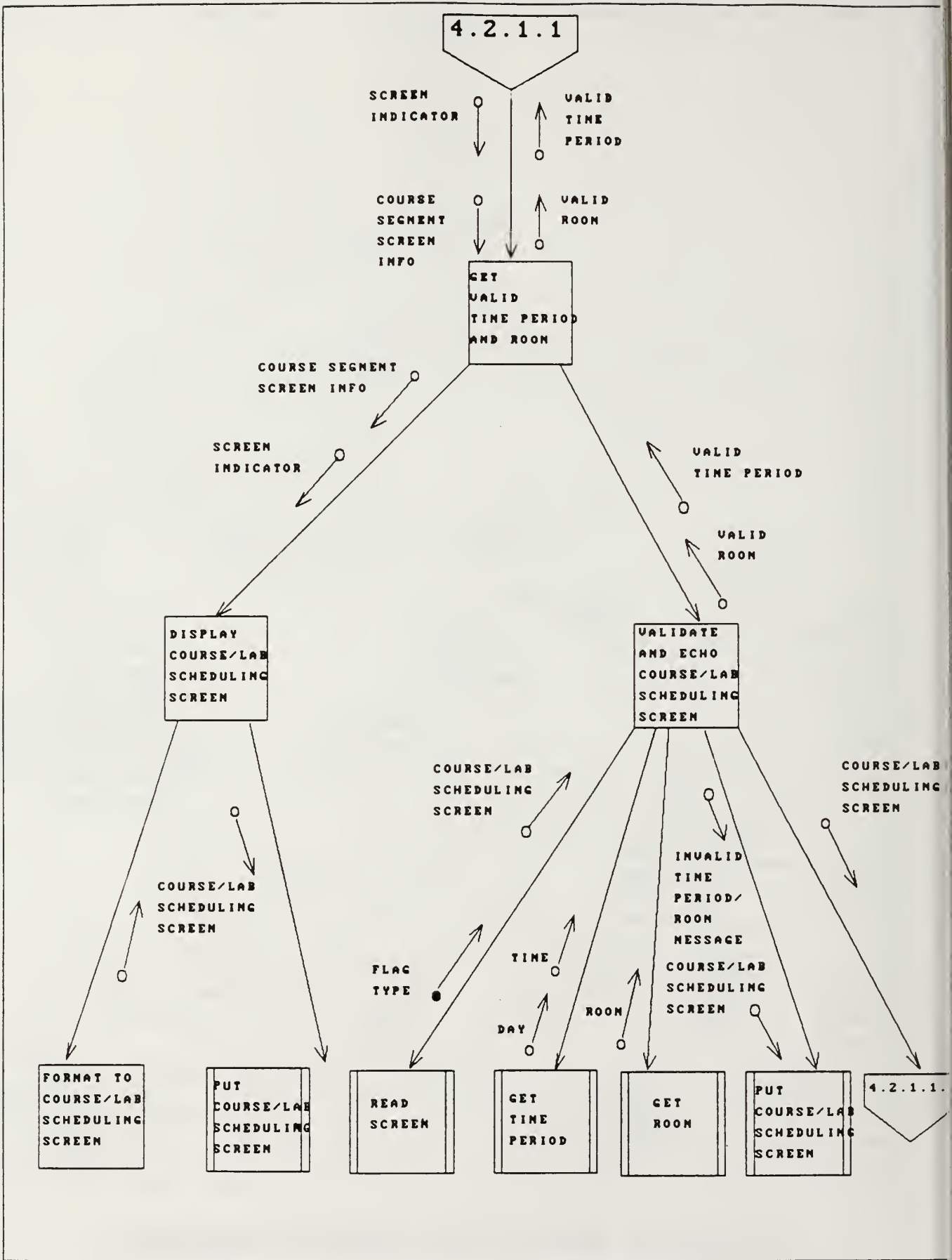


Fig. D-20, GET VALID TIME PERIOD AND ROOM.

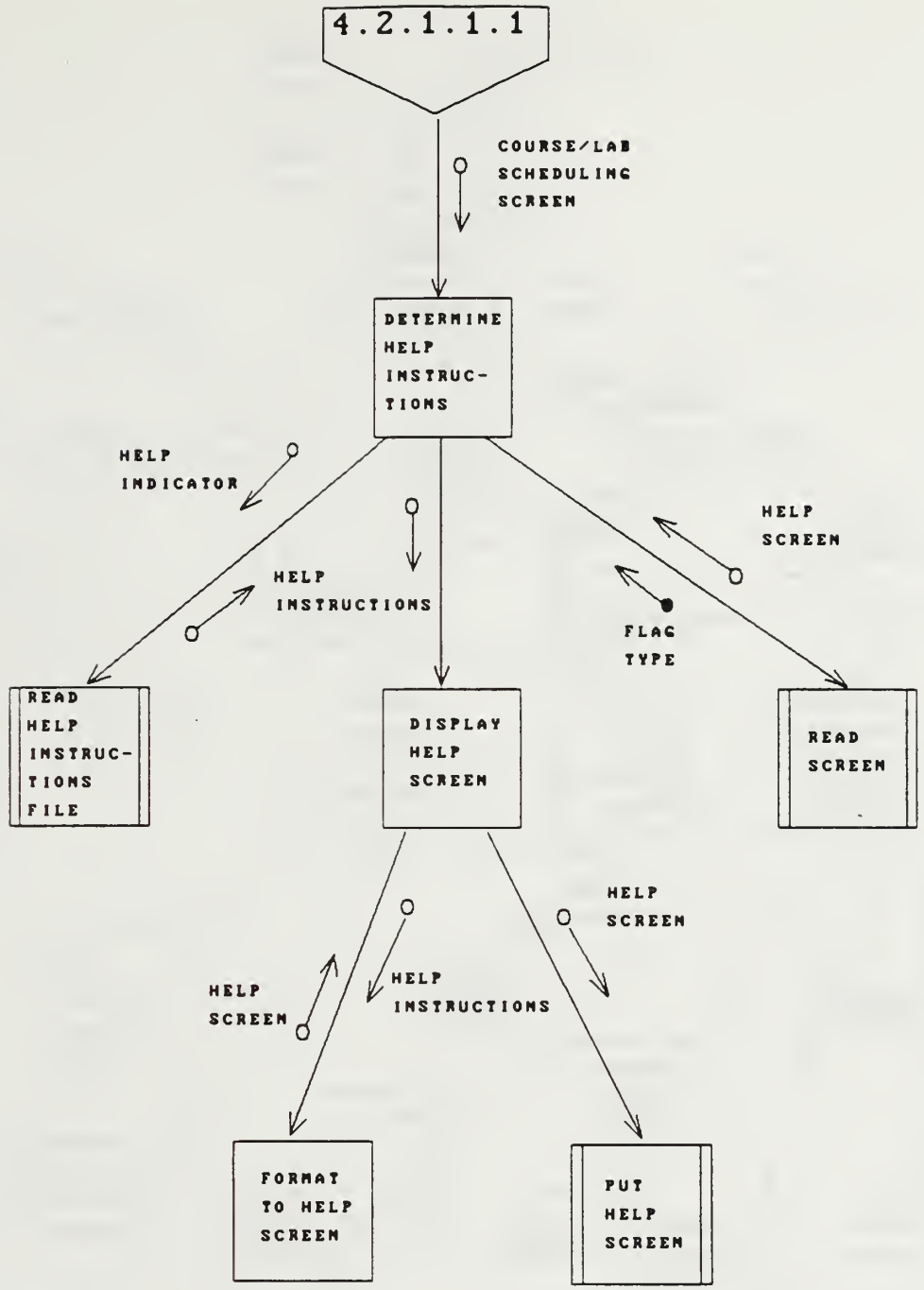


Fig. D-21, DETERMINE HELP INSTRUCTIONS for Course/Lab Scheduling Screen.

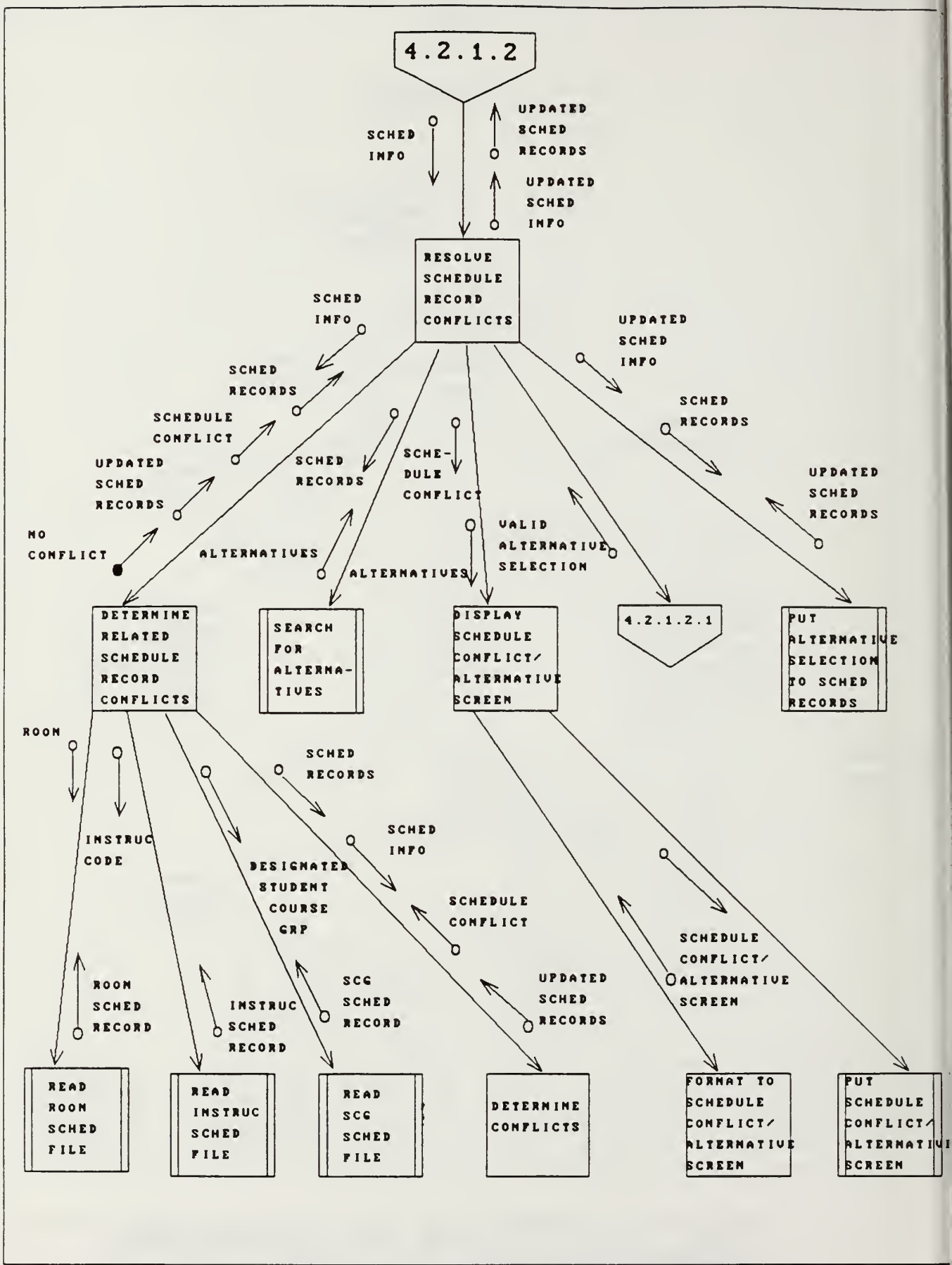


Fig. D-22, RESOLVE SCHEDULE RECORD CONFLICTS.

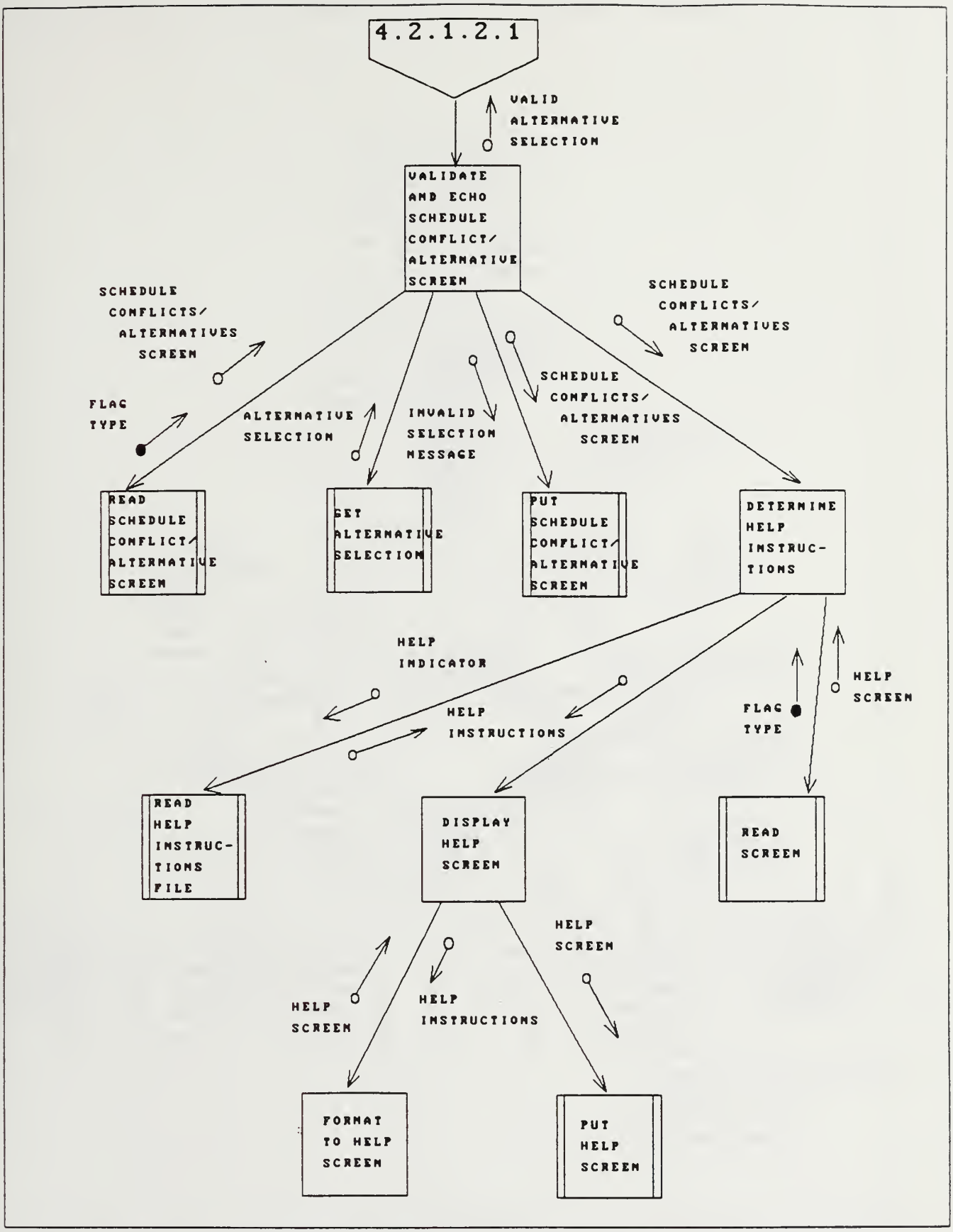


Fig. D-23, VALIDATE AND ECHO SCHEDULE CONFLICT/ALTERNATIVE SCREEN.

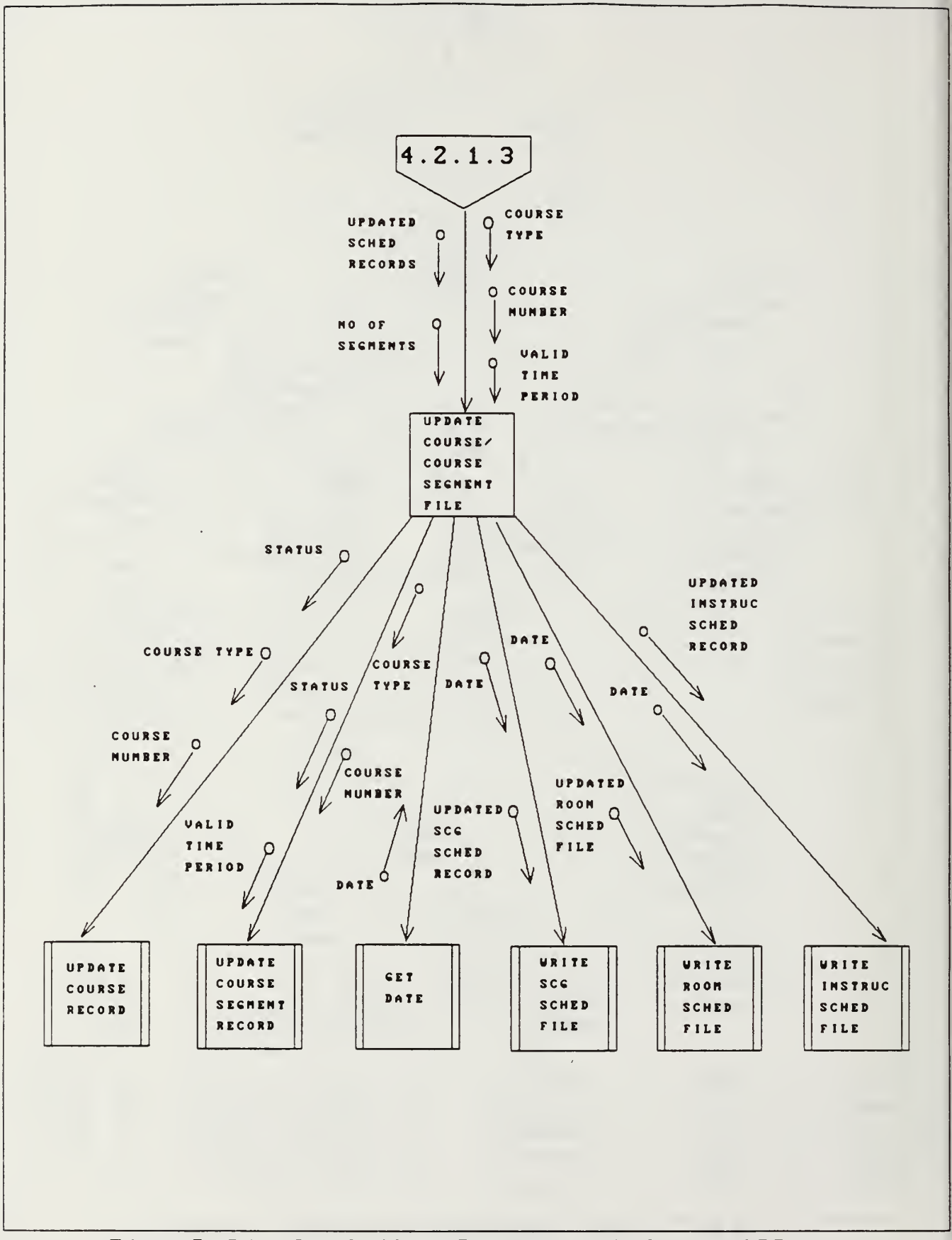


Fig. D-24, SAVE TO PERTINENT FILE, RECORD.

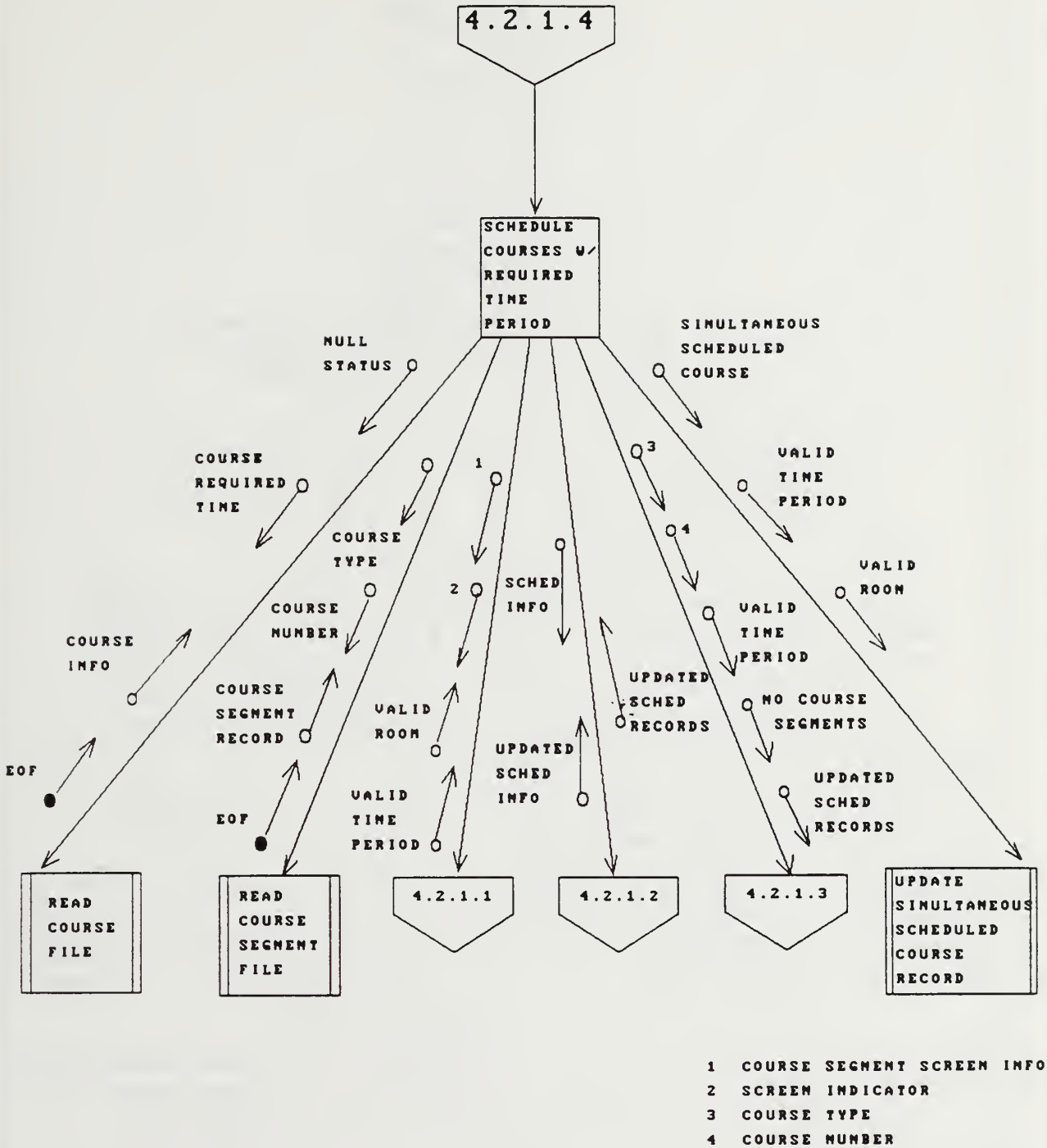


Fig. D-25, SCHEDULE COURSES W/ REQUIRED TIME PERIOD.

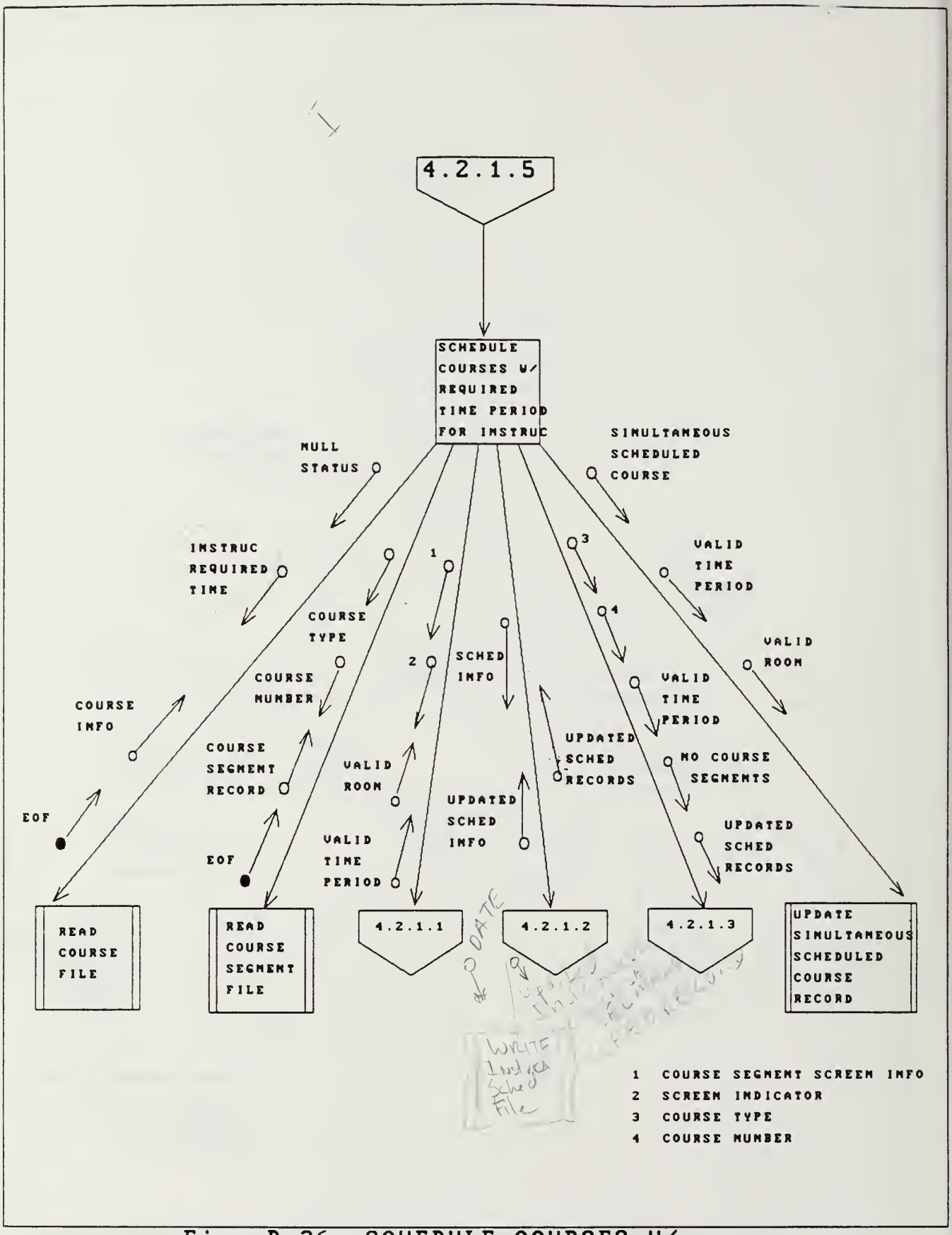


Fig. D-26, SCHEDULE COURSES W/ REQUIRED TIME PERIOD FOR INSTRUC.

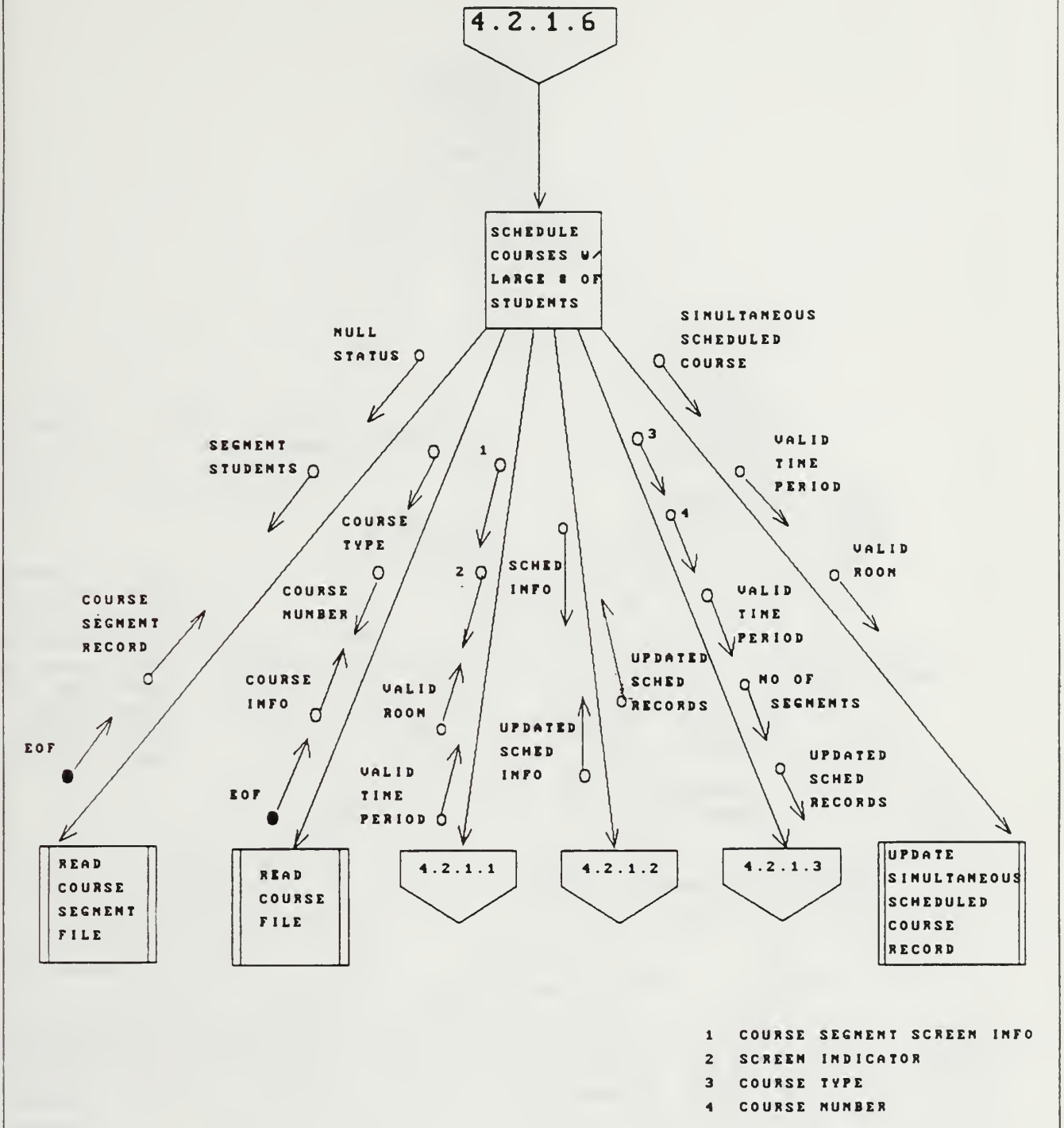


Fig. D-27, SCHEDULE COURSES W/ LARGE # OF STUDENTS.

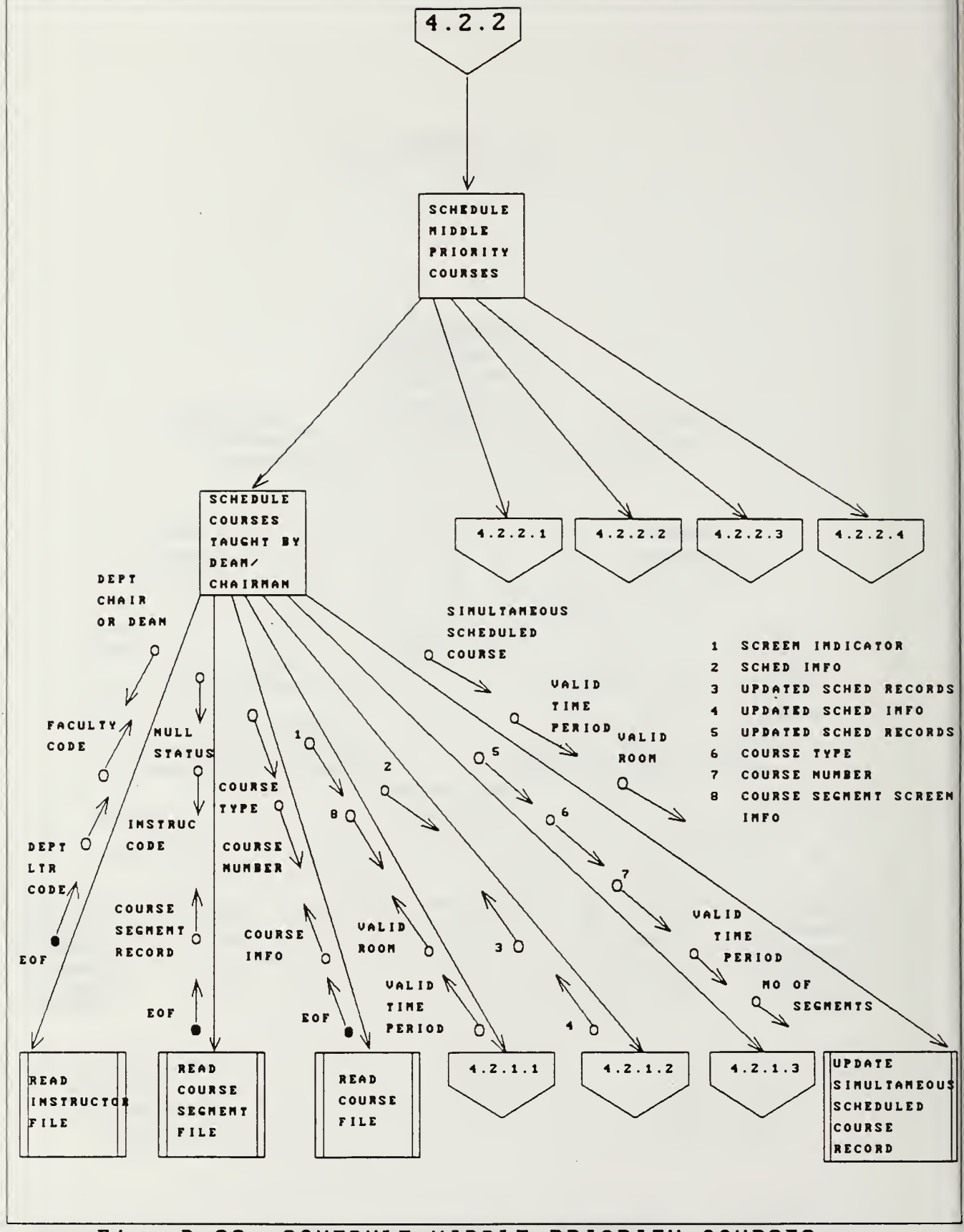


Fig. D-28, SCHEDULE MIDDLE PRIORITY COURSES.

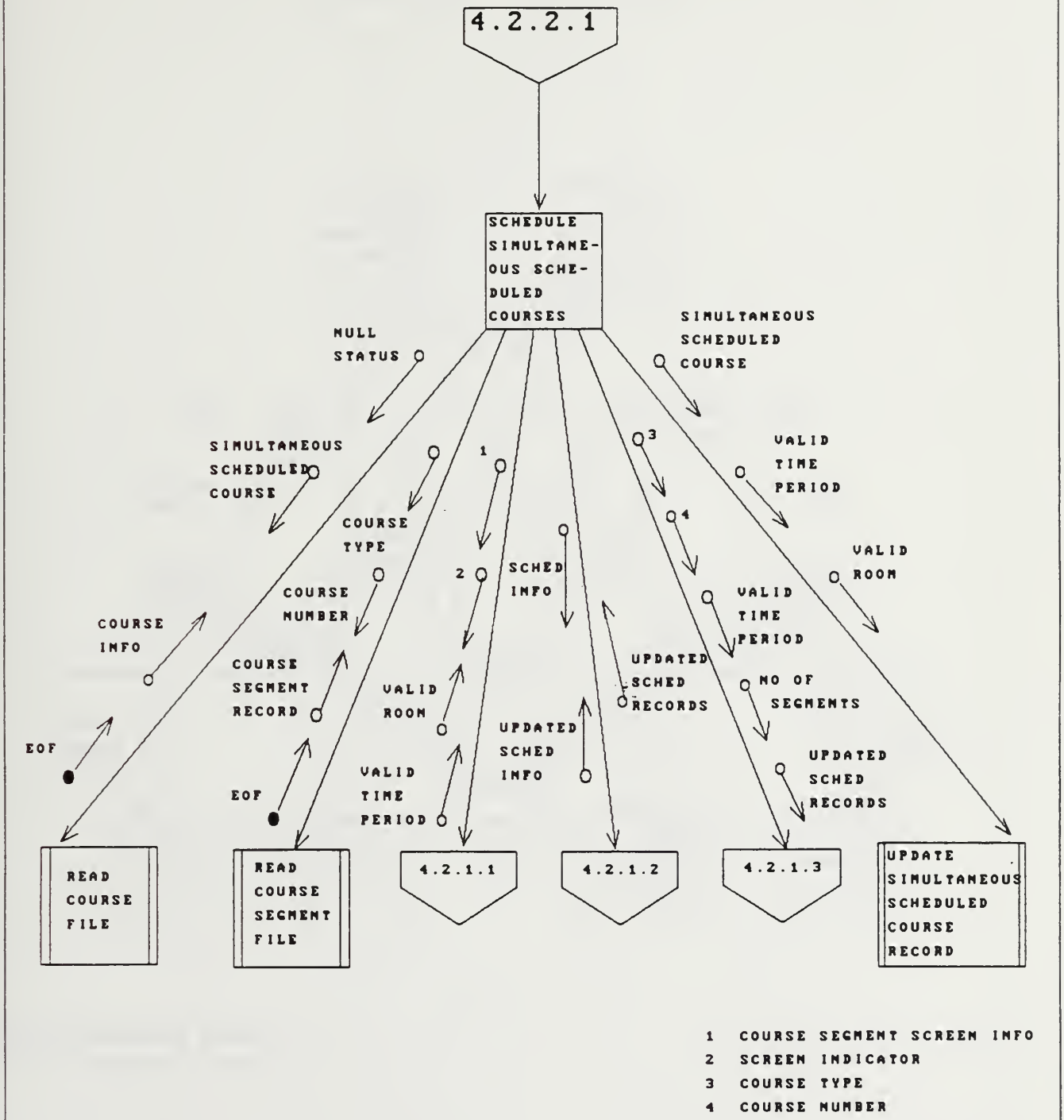


Fig. D-29, SCHEDULE SIMULTANEOUS SCHEDULED COURSES.

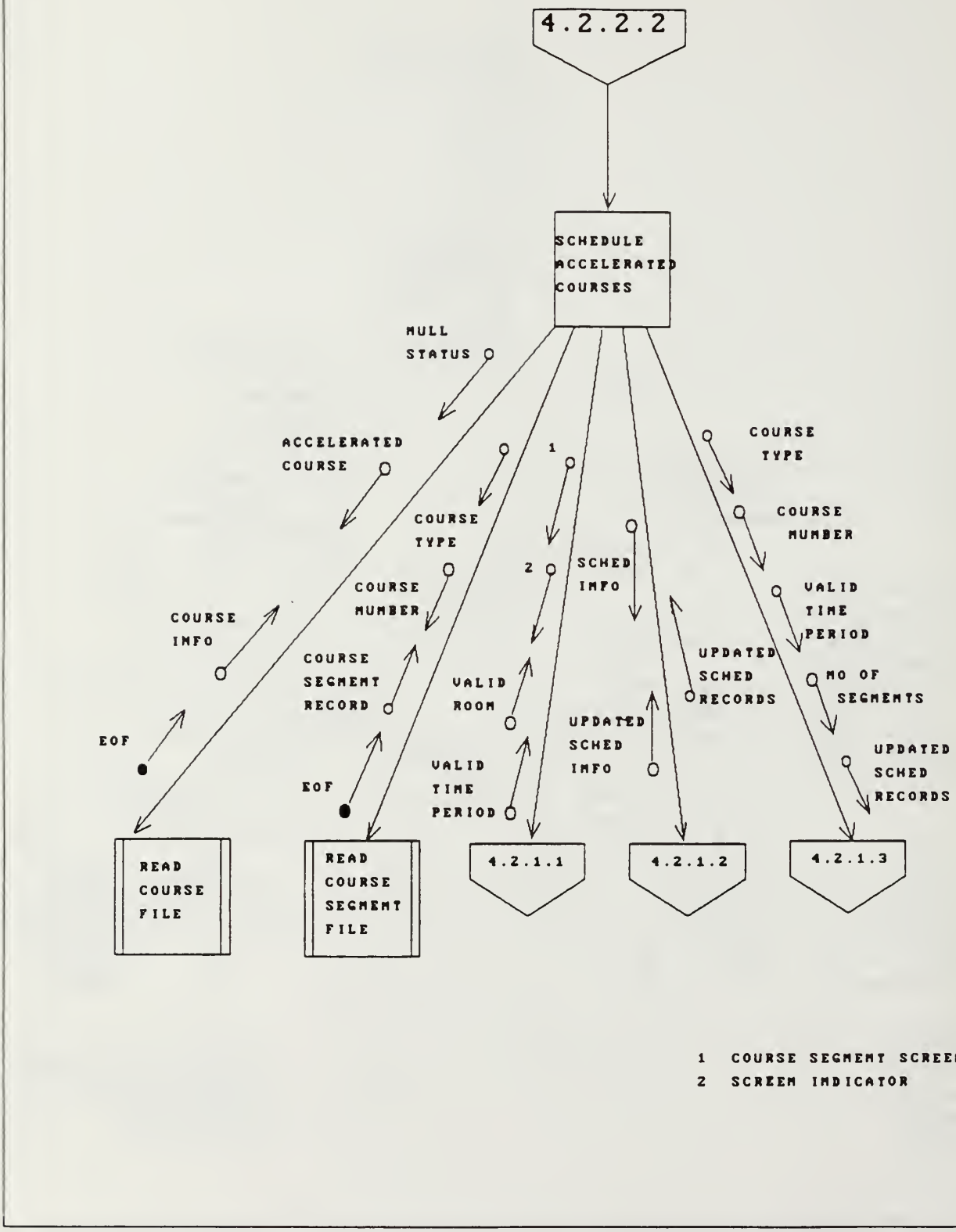


Fig. D-30, SCHEDULE ACCELERATED COURSES.

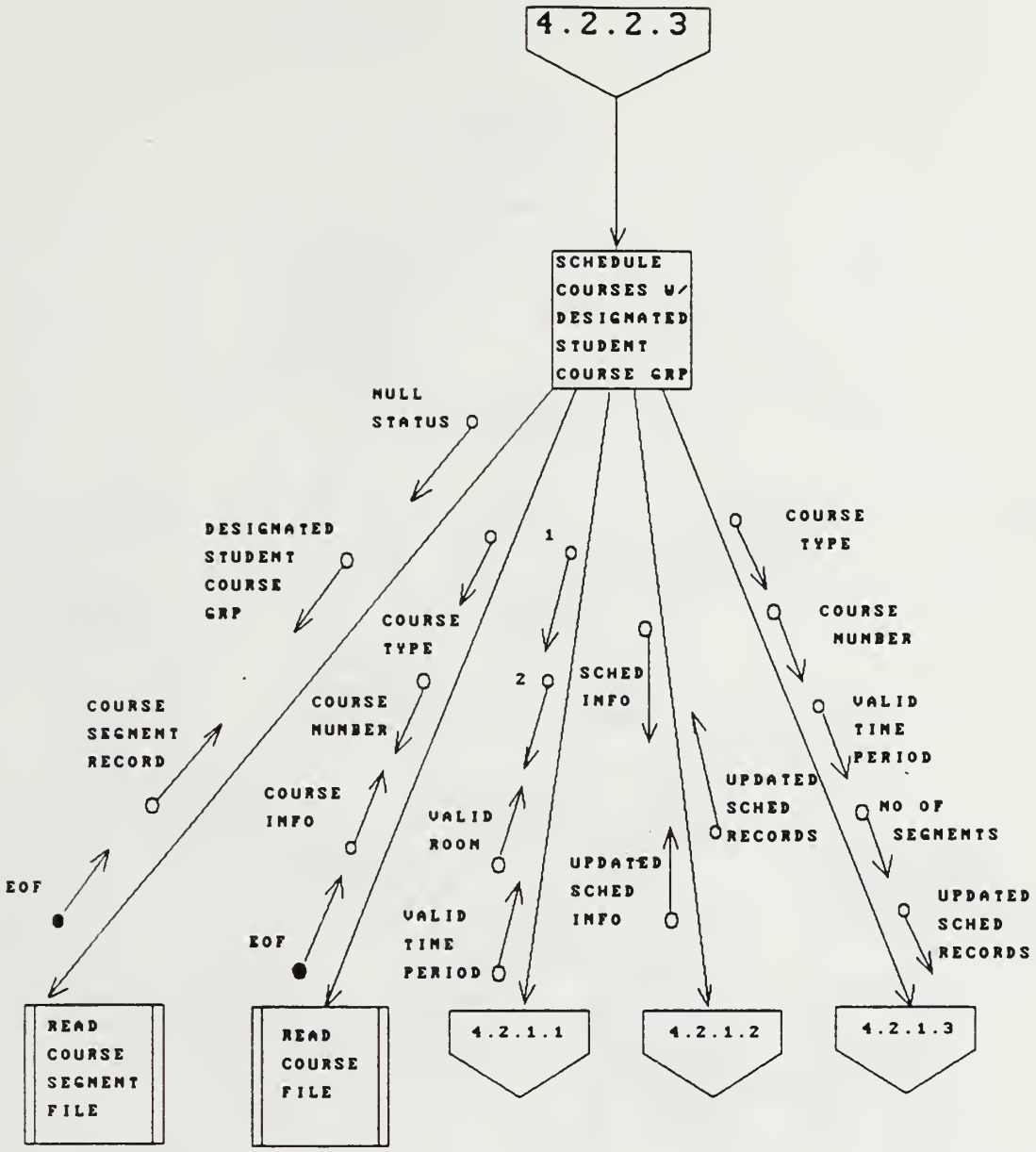


Fig. D-31, SCHEDULE COURSES W/ DESIGNATED STUDENT COURSE GRP.

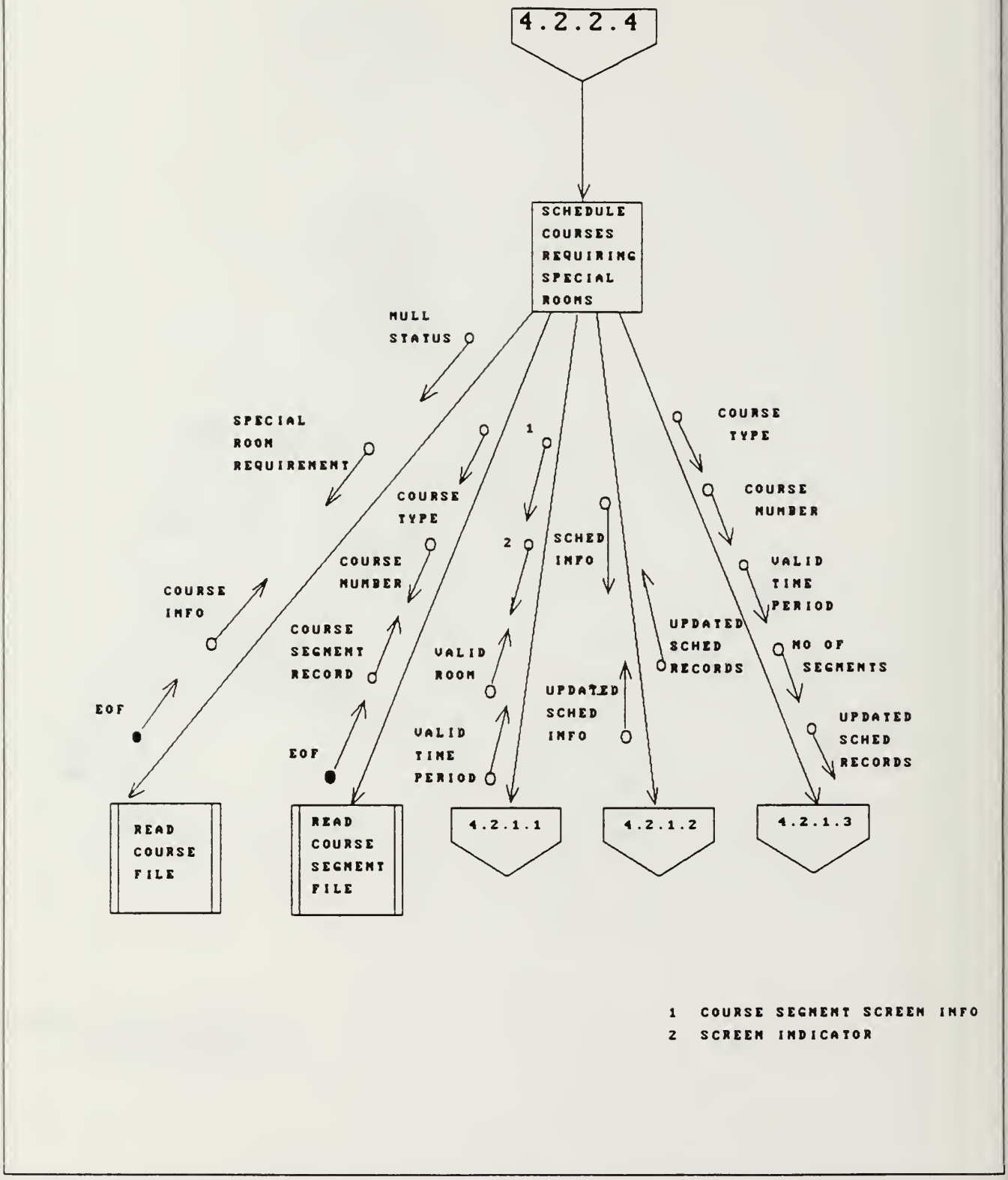


Fig. D-32, SCHEDULE COURSES REQUIRING SPECIAL ROOMS.

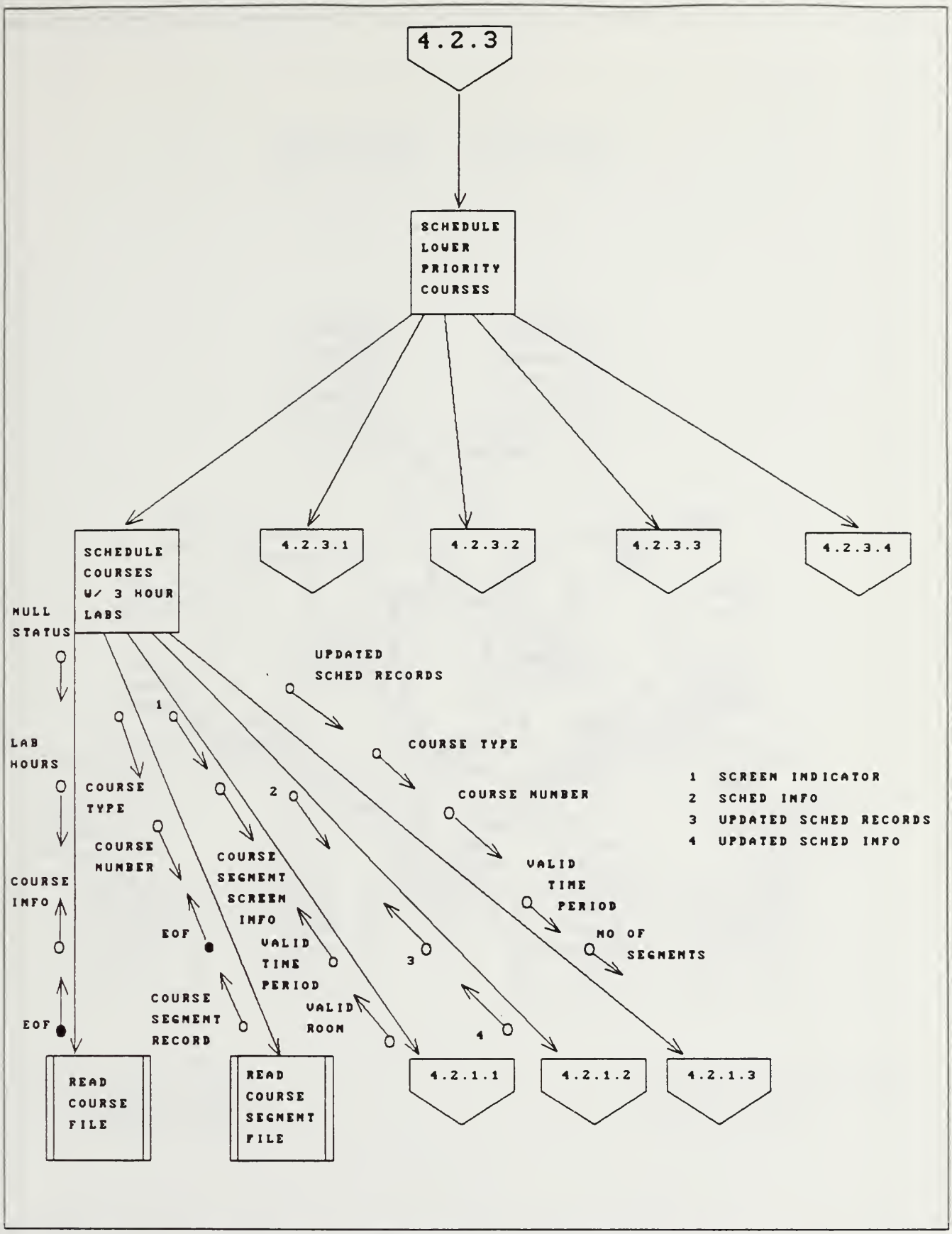


Fig. D-33, SCHEDULE LOWER PRIORITY COURSES.

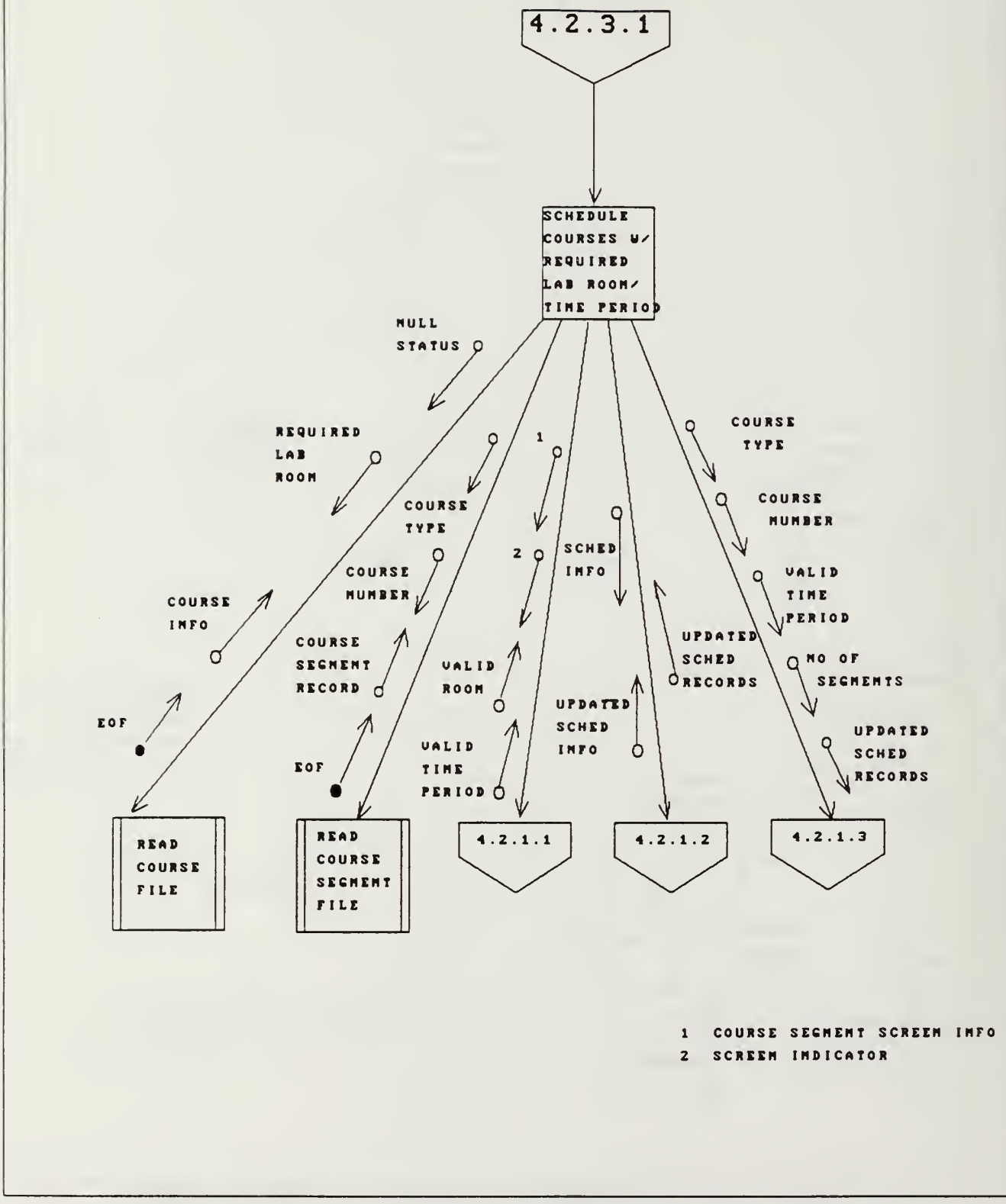


Fig. D-34, SCHEDULE COURSES W/ REQUIRED LAB ROOM/TIME PERIOD.

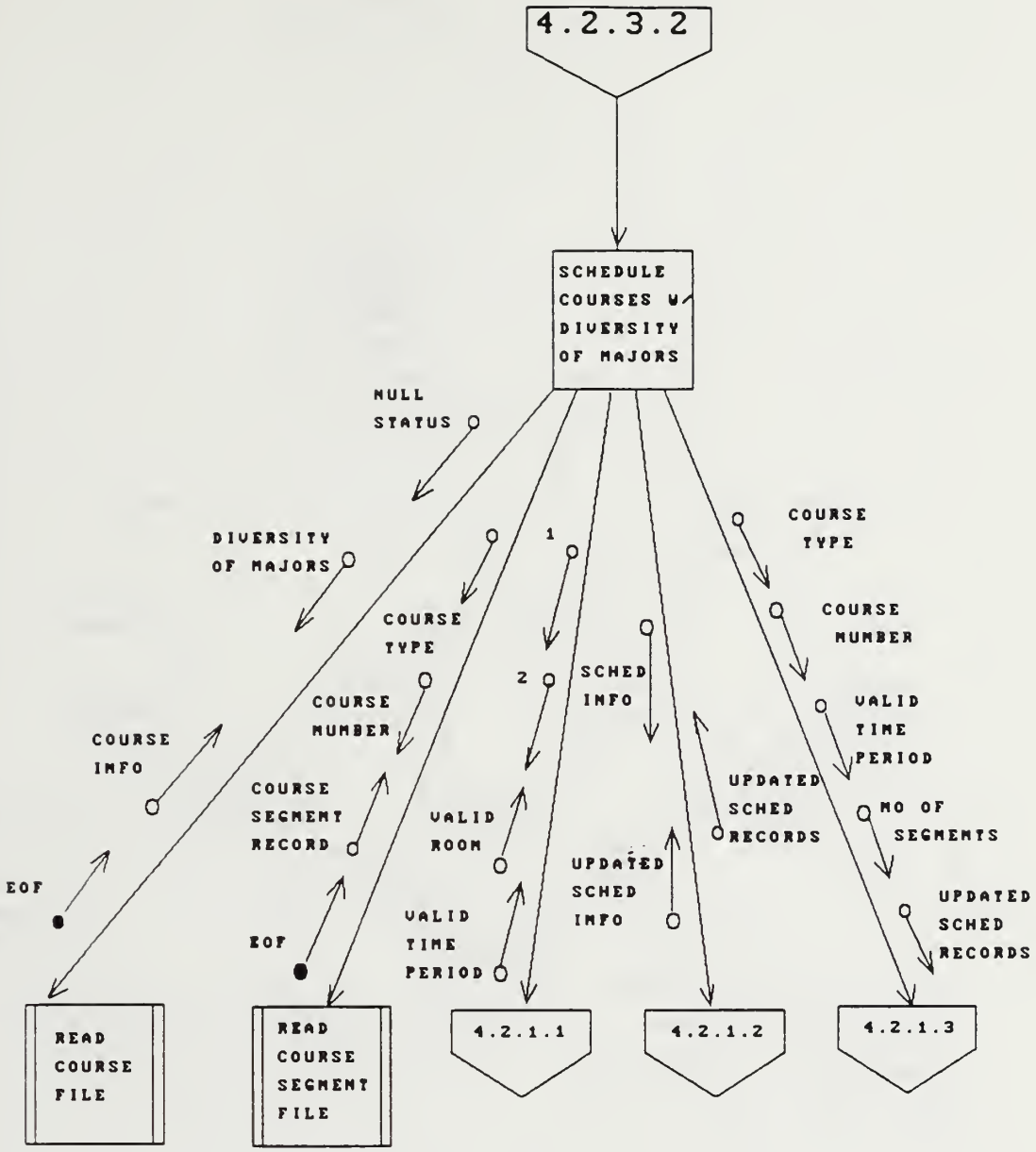


Fig. D-35, SCHEDULE COURSES W/ DIVERSITY OF MAJORS.

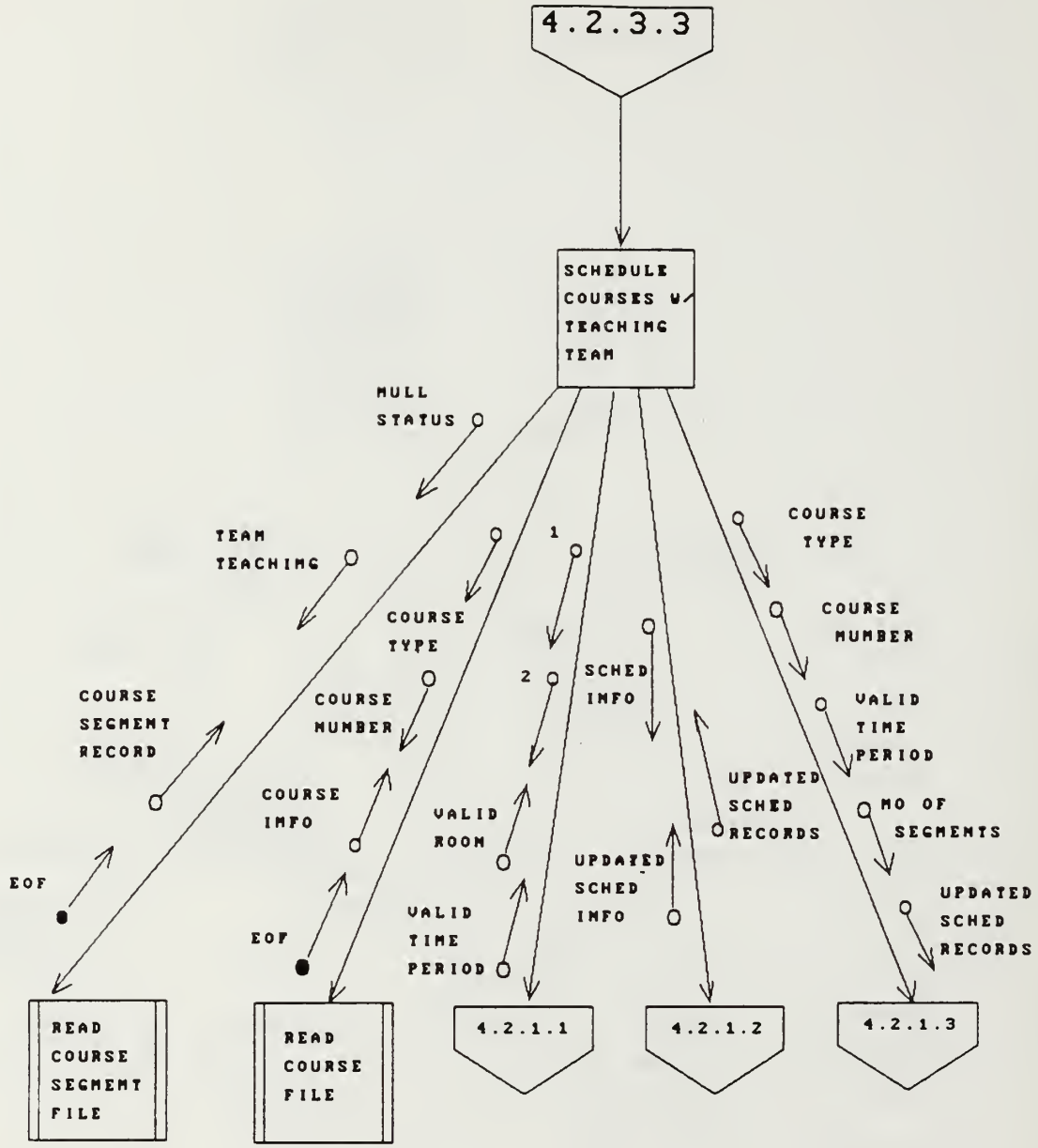


Fig. D-36, SCHEDULE COURSES W/ TEAM TEACHING.

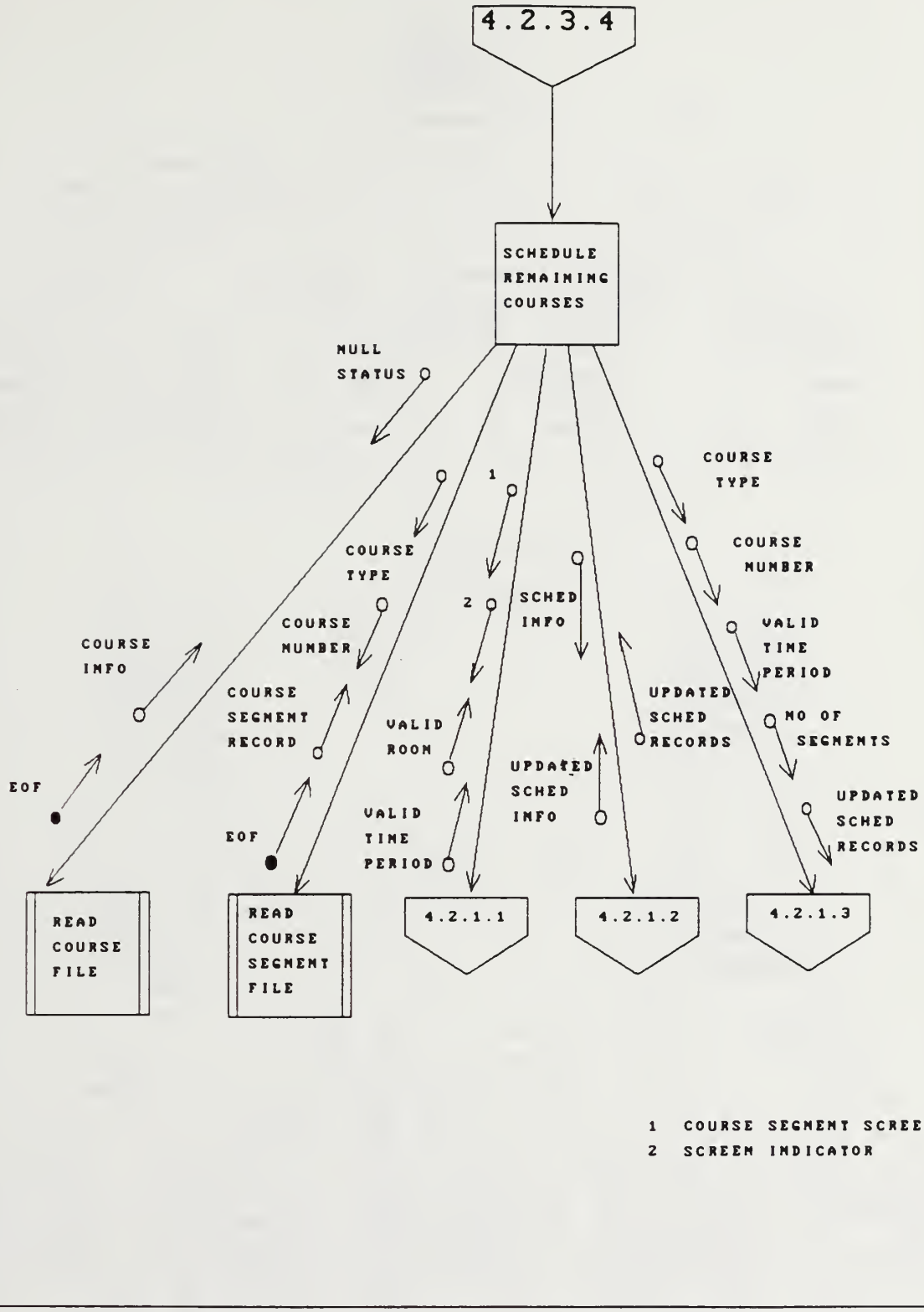


Fig. D-37, SCHEDULE REMAINING COURSES.

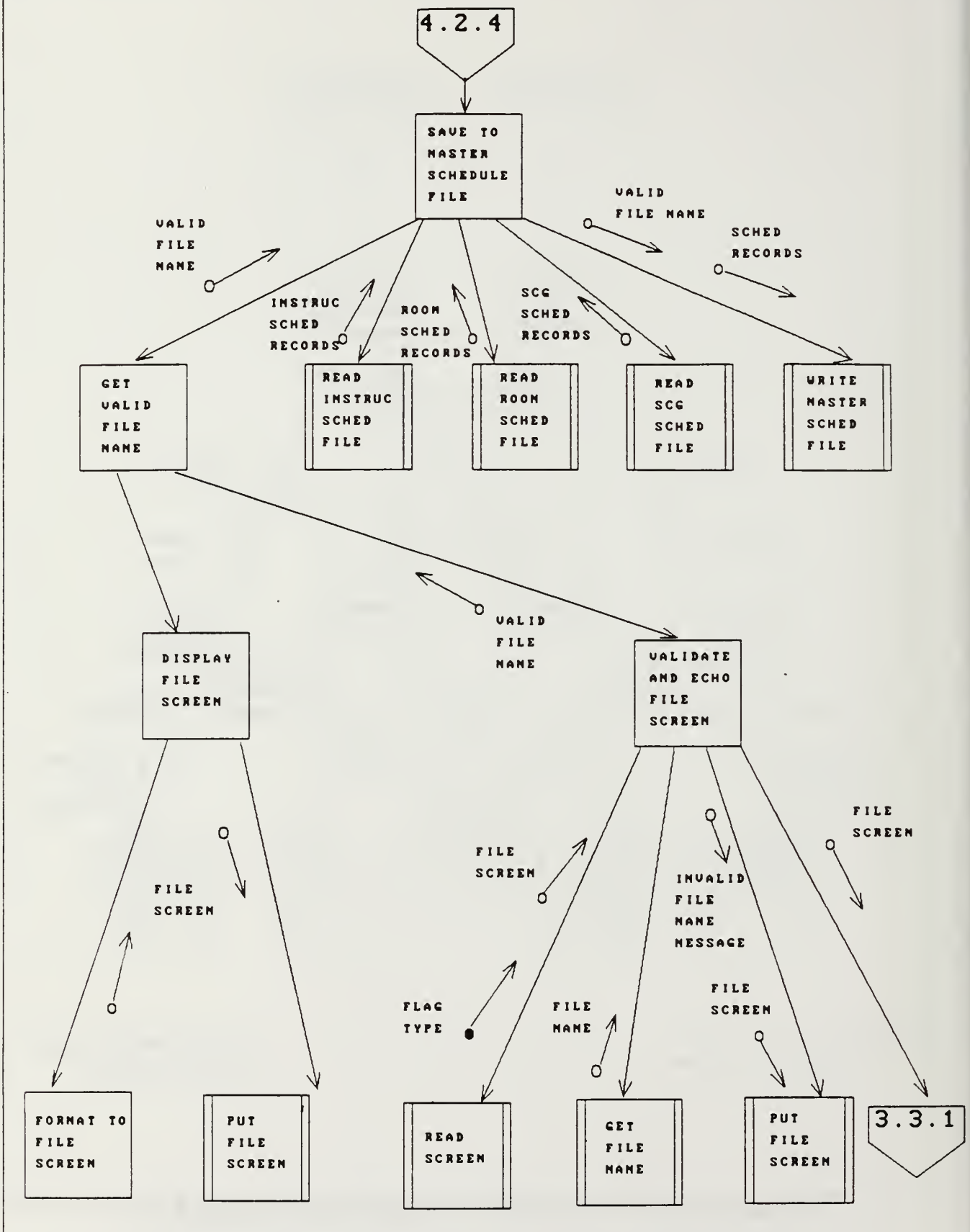


Fig. D-38, SAVE TO MASTER SCHEDULE FILE.

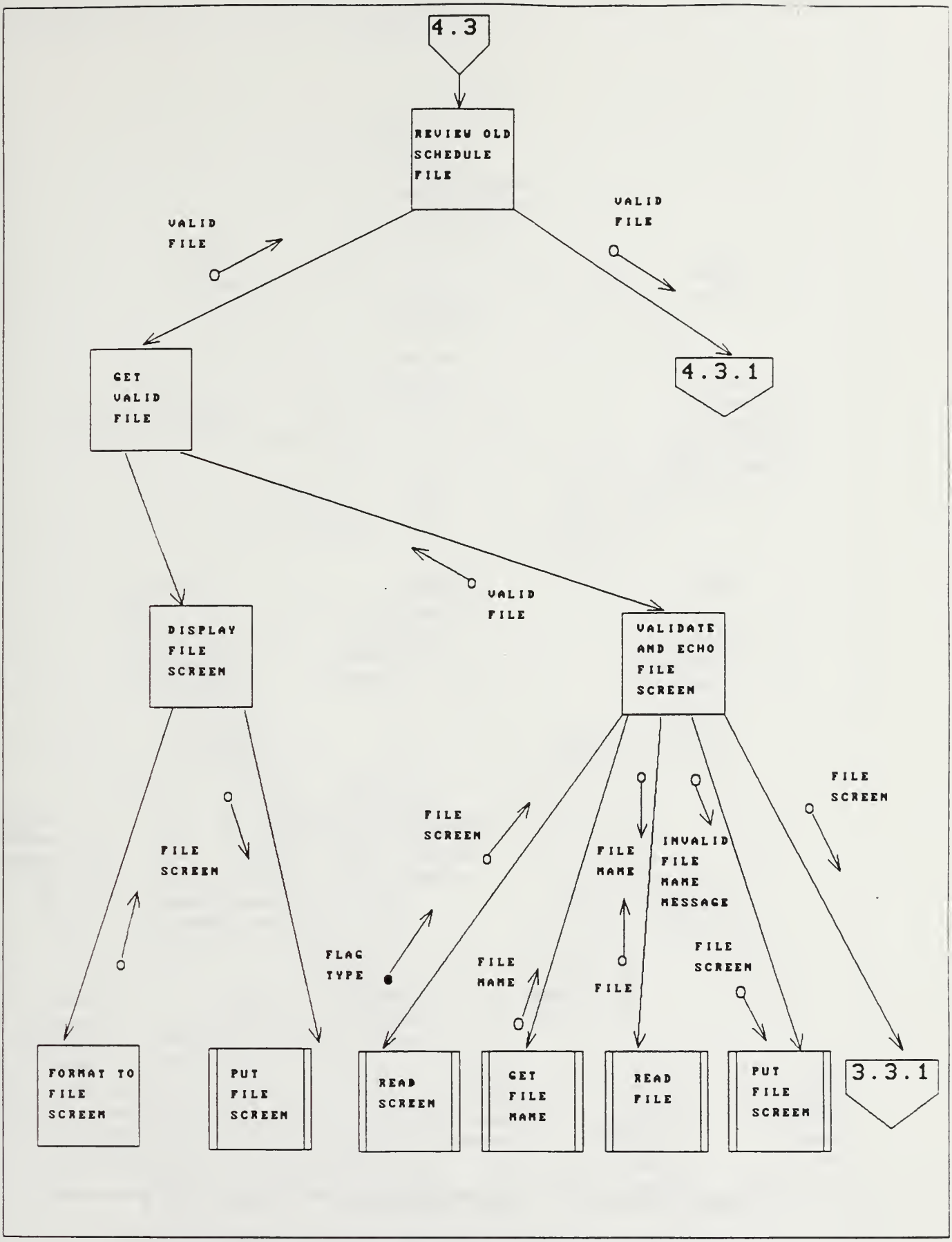


Fig. D-39, REVIEW OLD SCHEDULE FILE.

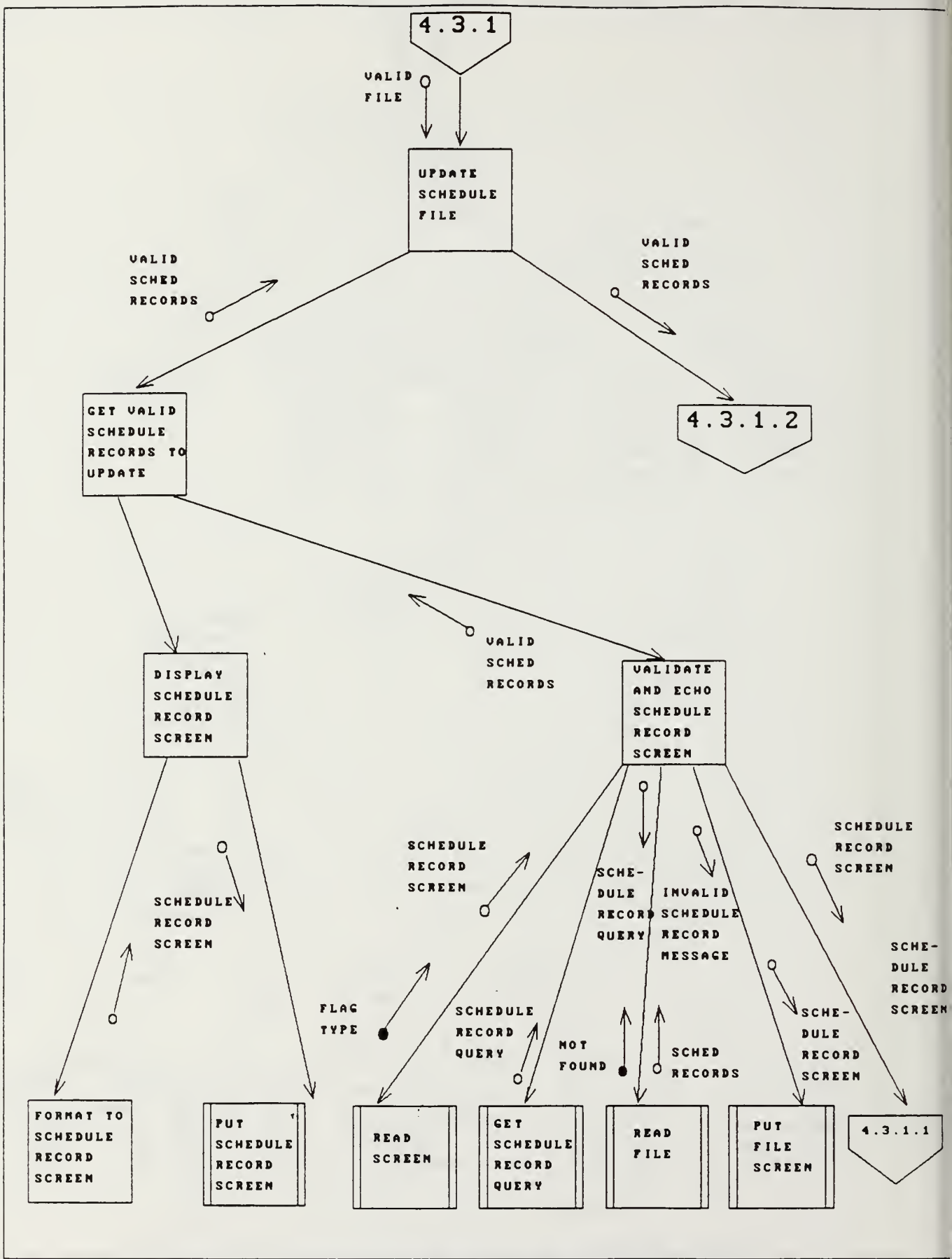


Fig. D-40, UPDATE SCHEDULE FILE.

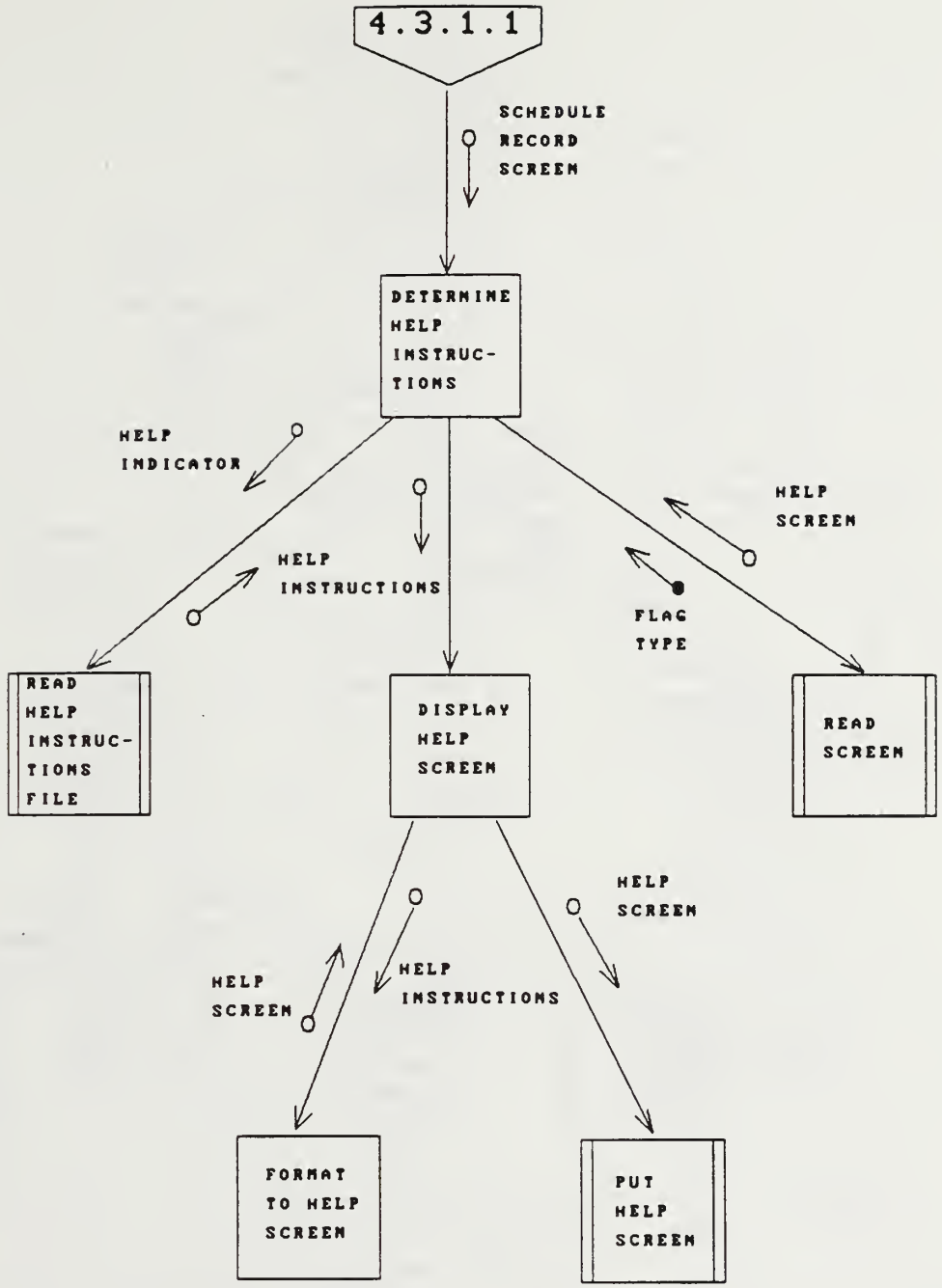


Fig. D-41, DETERMINE HELP INSTRUCTIONS for Schedule Record Screen.

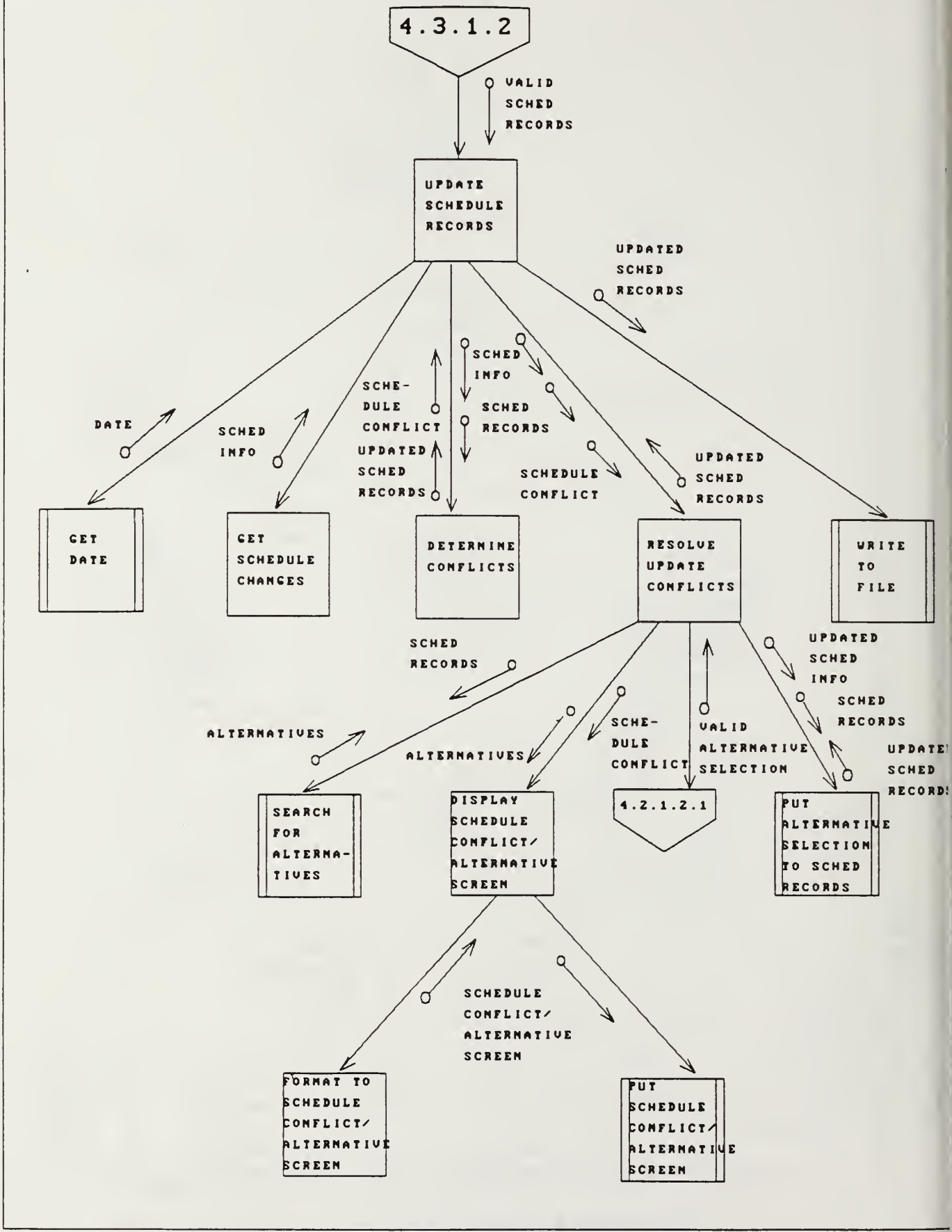


Fig. D-42, UPDATE SCHEDULE RECORDS.

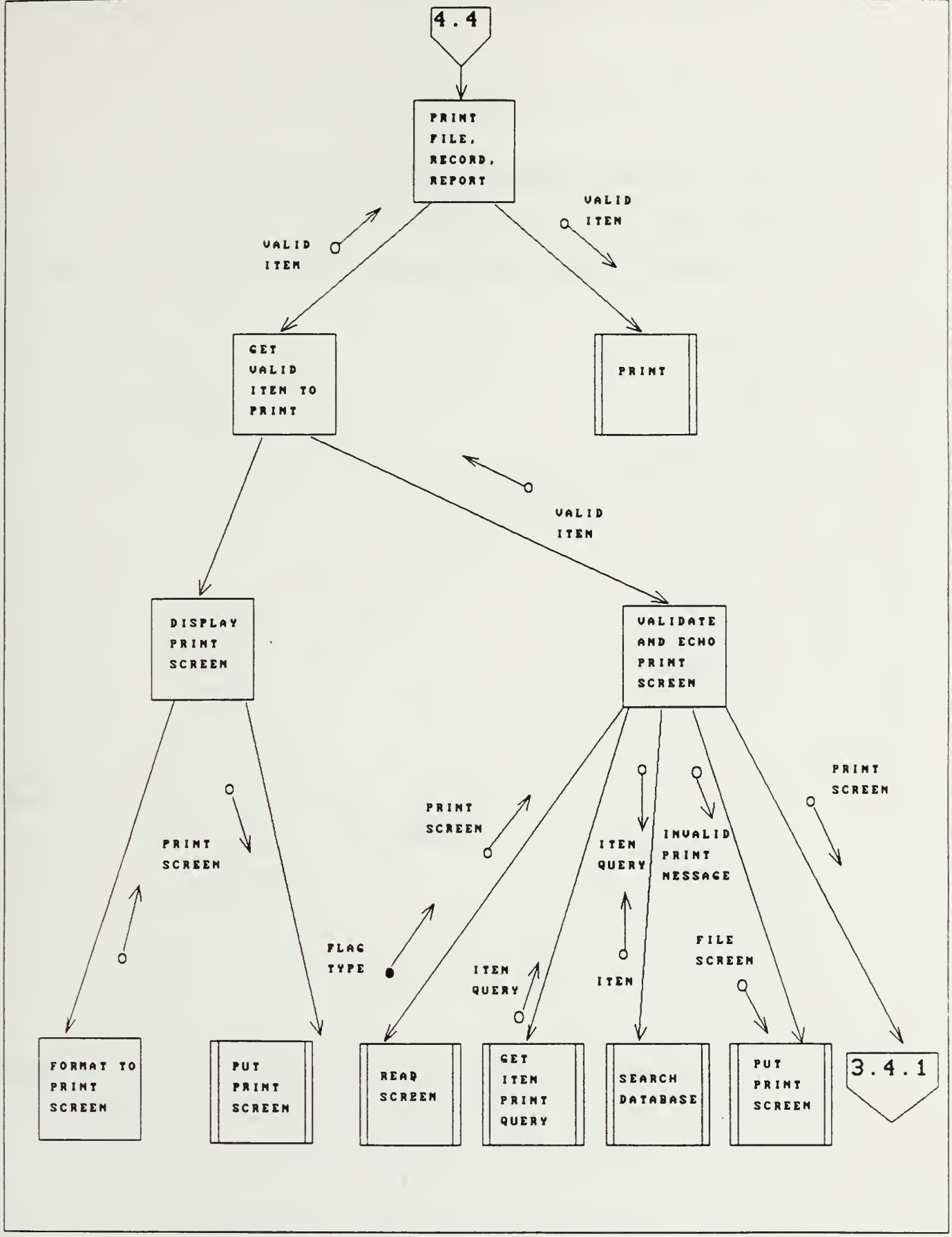


Fig. D-43, PRINT FILE, RECORD, REPORT.

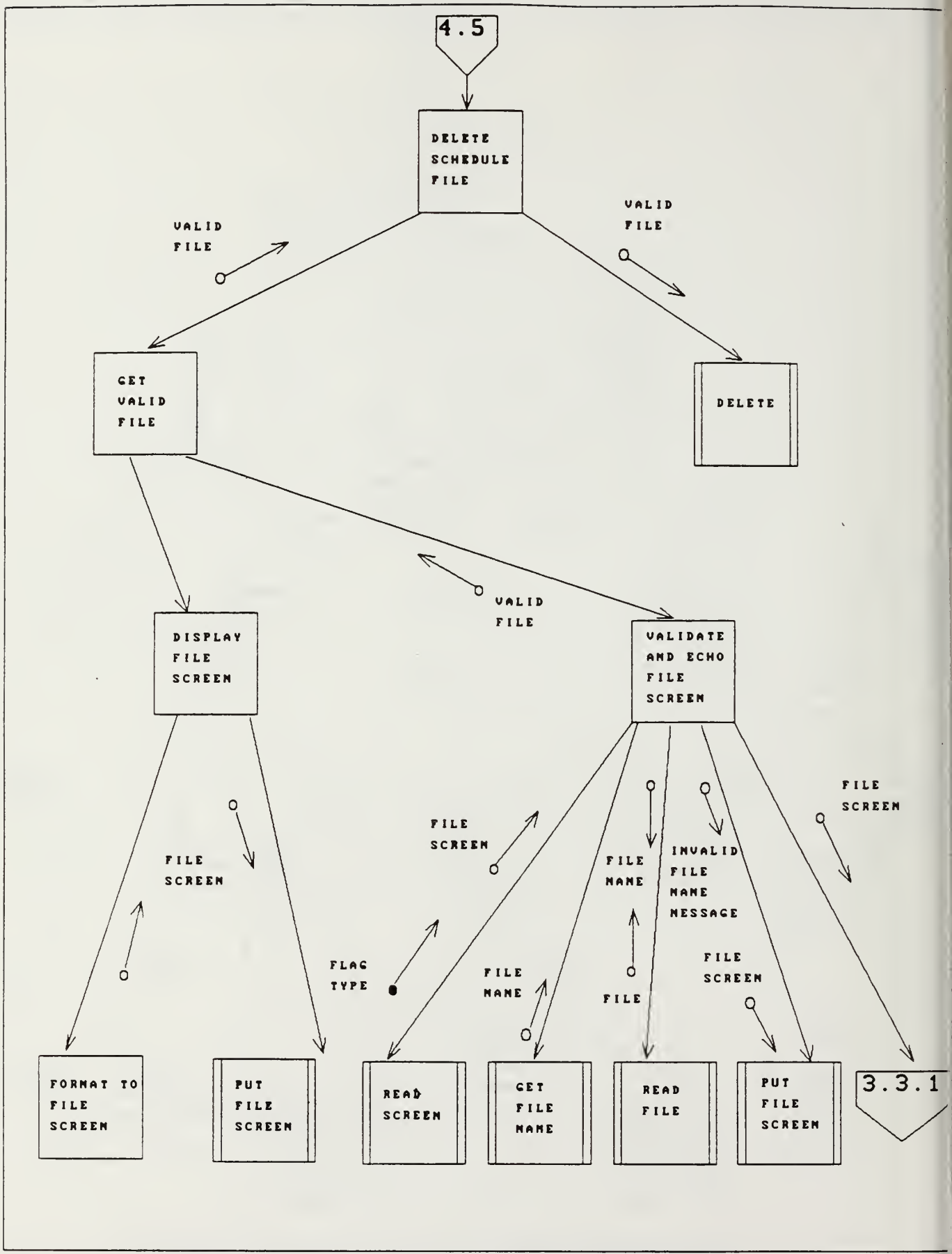


Fig. D-44, DELETE SCHEDULE FILE.

APPENDIX E: MODULE SPECIFICATIONS

The following report details, in alphabetical order, the specifications of each module illustrated in Appendix D.

MODULE: ADD DATA RECORD

PURPOSE: To add a data record to the given file.

USES: VALID FILE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Add a record to the VALID FILE.

CALLED BY: UPDATE DATABASE FILES

MODULE: ASSIGN SEMI-PERM EVENT TO SCHEDULES

PURPOSE: To assign the SEMI-PERM EVENT and its TIME PERIOD and ROOM to the related SCHED RECORDS.

USES: SCHED RECORDS
SEMI-PERM EVENT RECORDS

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  SCHED RECORDS.TIME PERIOD :=  SEMI-PERM EVENT
                                  RECORD.TIME PERIOD
  PUT SEMI-PERM EVENT RECORD in the SCHED RECORDS.
                                      TIME PERIOD slot
  Call WRITE SCG SCHED FILE (UPDATED SCG SCHED
                                                RECORD)
  Call WRITE ROOM SCHED FILE (UPDATED ROOM SCHED
                                                RECORD)
  Call WRITE INSTRUC SCHED FILE (UPDATED INSTRUC
                                              SCHEDRECORD)
End
```

CALLED BY: SCHEDULE SEMI-PERM EVENTS

MODULE: COPY MAINFRAME FILES

PURPOSE: To coordinate copying files downloaded form the NPS mainframe computer.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call UPDATE STUDENT COURSE GRP FILE
  Call UPDATE STUDENT FILE
  Call UPDATE COURSE/COURSE SEGMENT FILES
  Call UPDATE FACULTY FILE
End
```

CALLED BY: SCHEDULING DATA MAINTENANCE

MODULE: CREATE NEW SCHEDULES

PURPOSE: To coordinate creation of new event/course schedules based on priorities.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call SCHEDULE SEMI-PERM EVENTS
  Call SCHEDULE TOP PRIORITY COURSES
  Call SCHEDULE MIDDLE PRIORITY COURSES
  Call SCHEDULE LOWER PRIORITY COURSES
  Call SAVE TO MASTER SCHEDULE FILE
End
```

CALLED BY: MANAGE EVENT/COURSE SCHEDULING

MODULE: DELETE

PURPOSE: To delete a given schedule file.

USES: VALID FILE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Delete the VALID FILE.

CALLED BY: DELETE SCHEDULE FILE

MODULE: DELETE DATA RECORD

PURPOSE: To delete a data record in the given file.

USES: VALID FILE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Delete a record in the VALID FILE.

CALLED BY: UPDATE DATABASE FILES

MODULE: DELETE SCHEDULE FILE

PURPOSE: To coordinate deletion of a schedule file.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call GET VALID FILE
  Call DELETE (VALID FILE)
End
```

CALLED BY: MANAGE EVENT/COURSE SCHEDULING

MODULE: DETERMINE CONFLICTS

PURPOSE: To put the schedule information to the related SCHEDULE RECORDS.

USES: SCHED RECORDS
SCHED INFO

RETURNS: UPDATED SCHED RECORDS
SCHEDULE CONFLICT

FUNCTIONAL DETAILS:

```
Begin
  If INSTRUC SCHED RECORD.TIME PERIOD AND ROOM
              SCHED.TIME PERIOD AND STUDENT COURSE
                                  GRP.TIME PERIOD = null
    For INSTRUC SCHED RECORD
      Put (COURSE TYPE, COURSE NUMBER,SEGMENT
            NUMBER, SEGMENT STUDENTS, VALID ROOM)
               in TIME PERIOD slot
    For ROOM SCHED RECORD
      Put (COURSE TYPE, COURSE NUMBER, SEGMENT
         NUMBER, SEGMENT STUDENTS, INSTRUC CODE)
              in TIME PERIOD slot
    For STUDENT COURSE GRP RECORD
      Put (COURSE TYPE, COURSE NUMBER, SEGMENT
              NUMBER VALID ROOM, INSTRUC CODE)
               in TIME PERIOD slot
    Return (UPDATED SCHED RECORDS)
  Else Return (SCHED CONFLICT)
  End if
End
```

CALLED BY: DETERMINE RELATED SCHEDULE RECORD CONFLICTS
UPDATE SCHEDULE RECORDS

MODULE: DETERMINE HELP INSTRUCTIONS

PURPOSE: To determine the help instuctions needed for a given screen.

USES:
PASSWORD SCREEN
TRANSACTION SCREEN
FILE SCREEN
PRINT SCREEN
COURSE/LAB SCHEDULING SCREEN
SCHEDULE CONFLICT/ALTERNATIVES SCREEN
SCHEDULE RECORD SCREEN

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Case Screen
    When Screen = COURSE/LAB SCHEDULING SCREEN
      Call READ HELP INSTRUCTIONS FILE (COURSE/LAB
                                  HELP SCREEN INDICATOR)
    When Screen = SCHEDULE CONFLICT/ALTERNATIVES
                                  SCREEN
      Call READ HELP INSTRUCTIONS FILE
                          (CONFLICT/ALTERNATIVESHELP
                                  SCREEN INDICATOR)
    When Screen = PASSWORD SCREEN
      Call READ HELP INSTRUCTIONS FILE (PASSWORD
                                  HELP SCREEN INDICATOR)
    When Screen = TRANSACTION SCREEN
      Call READ HELP INSTRUCTIONS FILE (TRANSACTION
                                  HELP SCREEN INDICATOR)
    When Screen = FILE SCREEN
      Call READ HELP INSTRUCTIONS FILE (FILE HELP
                                  SCREEN INDICATOR)
    When Screen = PRINT SCREEN
      Call READ HELP INSTRUCTIONS FILE (PRINT HELP
                                  SCREEN INDICATOR)
    When Screen = SCHEDULE RECORD SCREEN
      Call READ HELP INSTRUCTIONS FILE (SCHEDULE
                          RECORD HELP SCREEN INDICATOR)
  Endcase
  Call DISPLAY HELP SCREEN (HELP INSTRUCTONS)
  Call READ SCREEN
  If FLAG TYPE = RETURN
    Go back to where called
  Elseif FLAG TYPE = QUIT
    Exit system
  Endif
End
```

CALLED BY: VALIDATE AND ECHO PASSWORD SCREEN
VALIDATE AND ECHO TRANSACTION SCREEN
VALIDATE AND ECHO FILE SCREEN
VALIDATE AND ECHO PRINT SCREEN
VALIDATE AND ECHO COURSE/LAB SCHEDULING SCREEN
VALIDATE AND ECHO SCHEDULE CONFLICT/ALTERNATIVES SCREEN
VALIDATE AND ECHO SCHEDULE RECORD SCREEN

MODULE: DETERMINE MBRSHP SCHEDS AND ROOM SCHED

PURPOSE: To determine the membership and room schedule records for a given semi-permanent event.

USES: MBRSHP
ROOM

RETURNS: SCHED RECORDS

FUNCTIONAL DETAILS:

```
Begin
  Call READ SCG SCHED FILE (DESIGNATED STUDENT
                            COURSE GRP) until EOF
  Call READ ROOM SCHED FILE (ROOM) until EOF
  Call READ INSTRUC SCHED FILE (INSTRUC CODE) until
                                                 EOF

  Return all related SCHED RECORDS
End
```

CALLED BY: SCHEDULE SEMI-PERM EVENTS

MODULE: DETERMINE RELATED SCHEDULE RECORD CONFLICTS

PURPOSE: To determine the SCHEDULE CONFLICTS that exist between the related SCHEDULE RECORDS.

USES: COURSE SEGMENT SCHED INFO

RETURNS: NO CONFLICT (flag)
UPDATED SCHED RECORDS
SCHEDULE CONFLICT

FUNCTIONAL DETAILS:

```
Begin
  Call READ ROOM SCHED FILE (ROOM)
  Call READ INSTRUC SCHED FILE (INSTRUC CODE)
  Call READ SCG SCHED FILE (DESIGNATED STUDENT
                                    COURSE GRP'S)
  Call DETERMINE COMFLICTS (SCHED RECORDS, SCHED
                                          INFO)
  If SCHEDULE CONFLICT exists
    Return (SCHED RECORDS, SCHEDULE CONFLICTS)
  Else
     Return (NO CONFLICT, UPDATED SCHED RECORDS)
  End if
End
```

CALLED BY: RESOLVE SCHEDULE RECORD CONFLICTS

MODULE: DETERMINE SEMI-PERM EVENT AND EVENT MBRSHP

PURPOSE: To determine a semi-permanent event and the membership of that event.

USES: N/A

RETURNS: MBRSHP
SEMI-PERM EVENT RECORD

FUNCTIONAL DETAILS:

```
Begin
  Call READ SEMI-PERM EVENT FILE
  If EOF go back to CREATE NEW SCHEDULE FILE
  Else CALL READ SEMI-PERM EVENT MBRSHP FILE (EVENT
                                        TYPE, EVENT NAME)
  End if
  Return (MBRSHP, SEMI-PERM EVENT RECORD)
End
```

CALLED BY: SCHEDULE SEMI-PERM EVENTS

MODULE: DISPLAY COURSE/LAB SCHEDULING SCREEN

PURPOSE: To display a CRT screen for TIME PERIOD and ROOM entry.

USES: COURSE SEGMENT SCREEN INFO
SCREEN INDICATOR

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call FORMAT TO COURSE/LAB SCHEDULING SCREEN
  Put COURSE SEGMENT SCREEN INFO and SCREEN
     INDICATOR to COURSE/LAB SCHEDULING SCREEN
  Call PUT COURSE/LAB SCHEDULING SCREEN
End
```

CALLED BY: GET VALID TIME PERIOD AND ROOM

MODULE: DISPLAY FILE SCREEN

PURPOSE: To display a CRT screen for filename entry.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call FORMAT TO FILE SCREEN
  Call PUT FILE SCREEN (FILE SCREEN)
End
```

CALLED BY: GET VALID FILE
GET VALID FILE NAME

MODULE: DISPLAY HELP SCREEN

PURPOSE: To display CRT screen of HELP INSTRUCTIONS.

USES: HELP INSTRUCTIONS

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call FORMAT HELP SCREEN (HELP INSTRUCTIONS)
  Call PUT HELP SCREEN (HELP SCREEN)
End
```

CALLED BY: DETERMINE HELP INSTRUCTIONS

MODULE: DISPLAY PASSWORD SCREEN

PURPOSE: To display a CRT screen for password entry.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call FORMAT TO PASSWORD SCREEN
  Call PUT PASSWORD SCREEN (PASSWORD SCREEN)
End
```

CALLED BY: GET VALID PASSWORD

MODULE: DISPLAY PRINT SCREEN

PURPOSE: To display a CRT screen for selecting an item to print.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call FORMAT TO PRINT SCREEN
  Call PUT PRINT SCREEN (PRINT SCREEN)
End
```

CALLED BY: GET VALID ITEM TO PRINT

MODULE: DISPLAY SCHEDULE CONFLICT/ALTERNATIVES SCREEN

PURPOSE: To display a CRT screen for selection of an alternative TIME PERIOD.

USES: ALTERNATIVES
SCHEDULE CONFLICT

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call FORMAT TO SCHEDULE CONFLICT/ALTERRNATIVES
                                              SCREEN
  Put ALTERNATIVES and SCHEDULE COMFLICT to
            SCHEDULE CONFLICT/ALTERNATIVES SCREEN
  Call PUT SCHEDULE CONFLICT/ALTERNATIVES SCREEN
End
```

CALLED BY: RESOLVE SCHEDULE RECORD CONFLICT
RESOLVE UPDATE CONFLICTS

MODULE: DISPLAY SCHEDULE RECORD SCREEN

PURPOSE: To display a CRT screen for input by the system user of schedule of schedule records to get to update.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call FORMAT TO SCHEDULE RRECORD SCREEN
  Call PUT SCHEDULE RECORD SCREEN
End
```

CALLED BY: GET VALID SCHEDULE RECORDS TO UPDATE

MODULE: DISPLAY TRANSACTION SCREEN

PURPOSE: To display a CRT screen for TRANSACTION entry.

USES: NPS^3 TRANSACTIONS
DATA MAINTENANCE TRANSACTIONS
UPDATE TRANSACTIONS
EVENT/COURSE TRANSACTIONS

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
   Call FORMAT TO TRANSACTION SCREEN
   Call PUT TRANSACTION SCREEN (TRANSACTION SCREEN)
End
```

CALLED BY: GET VALID NPS SCHEDULING SYSTEM TRANSACTION
GET VALID DATA MAINTENANCE TRANSACTION
GET VALID EVENT/COURSE SCHEDULING TRANSACTION
GET VALID FILE UPDATE TRANSACTION

MODULE: FORMAT TO COURSE/LAB SCHEDULING SCREEN

PURPOSE: To format the screen for TIME PERIOD and ROOM entry.

USES: N/A

RETURNS: COURSE/LAB SCHEDULING SCREEN

FUNCTIONAL DETAILS:

1. Format screen for COURSE/LAB TIME PERIOD and ROOM entry.

CALLED BY: DISPLAY COURSE/LAB SCHEDULING SCREEN

MODULE: FORMAT TO FILE SCREEN

PURPOSE: To format the CRT screen for filename entry.

USES: N/A

RETURNS: FILE SCREEN

FUNCTIONAL DETAILS:

1. Format screen with filename entry data and prompt.

CALLED BY: DISPLAY FILE SCREEN

MODULE: FORMAT TO HELP SCREEN

PURPOSE: To format the CRT screen for help instructions.

USES: HELP INSTRUCTIONS

RETURNS: HELP SCREEN

FUNCTIONAL DETAILS:

1. Format screen with given HELP INSTRUCTIONS.

CALLED BY: DISPLAY HELP SCREEN

MODULE: FORMAT TO PASSWORD SCREEN

PURPOSE: To format the CRT screen for password entry.

USES: N/A

RETURNS: PASSWORD SCREEN

FUNCTIONAL DETAILS:

1. Format screen with password entry data and prompt.

CALLED BY: DISPLAY PASSWORD SCREEN

MODULE: FORMAT TO PRINT SCREEN

PURPOSE: To format the CRT screen for print item query.

USES: N/A

RETURNS: PRINT SCREEN

FUNCTIONAL DETAILS:

1. Format screen with item query data and prompt.

CALLED BY: DISPLAY PRINT SCREEN

MODULE: FORMAT TO SCHEDULE CONFLICT/ALTERNATIVE SCREEN

PURPOSE: To format the screen for ALTERNATIVE SELECTION.

USES: N/A

RETURNS: SCHEDULE CONFLICT/ALTERNATIVES SCREEN

FUNCTIONAL DETAILS:

1. Format screen for selection of displayed alternative TIME PERIODs.

CALLED BY: DISPLAY SCHEDULE CONFLICT/ALTERNATIVES SCREEN

MODULE: FORMAT TO SCHEDULE RECORD SCREEN

PURPOSE: To format the CRT screen for schedule record queries by system user.

USES: N/A

RETURNS: SCHEDULE RECORD SCREEN

FUNCTIONAL DETAILS:

1. Format screen with schedule record query dqtq and prompt.

CALLED BY: DISPLAY SCHEDULE RECORD SCREEN

MODULE: FORMAT TO TRANSACTION SCREEN

PURPOSE: To format the CRT screen for TRANSACTION entry.

USES: N/A

RETURNS: TRANSACTION SCREEN

FUNCTIONAL DETAILS:

1. Format screen with TRANSACTION entry data and prompt.

CALLED BY: DISPLAY TRANSACTION SCREEN

MODULE: GET ALTERNATIVE SELECTION

PURPOSE: To get an ALTERNATIVE SELECTION, from a list of ALTERNATIVES, entered by the system user.

USES: N/A

RETURNS: ALTERNATIVE SELECTION

FUNCTIONAL DETAILS: (Pre-defined)

1. Enter ALTERNATIVE SELECTION.

CALLED BY: VALIDATE AND ECHO SCHEDULE CONFLICT/ALTERNATIVES SCREEN

MODULE: GET DATE

PURPOSE: To get the current date from the operating system.

USES: N/A

RETURNS: DATE

FUNCTIONAL DETAILS: (Pre-defined)

1. Get current date from operating system.

CALLED BY: SAVE TO PERTINENT FILE/RECORD
UPDATE SCHEDULE RECORDS

MODULE: GET FILE NAME

PURPOSE: To get a FILENAME from system user.

USES: N/A

RETURNS: FILENAME

FUNCTIONAL DETAILS: (Pre-defined)

1. Enter FILENAME.

CALLED BY: VALIDATE AND ECHO FILE SCREEN

MODULE: GET ITEM PRINT QUERY

PURPOSE: To get a query on an item to print from the system user.

USES: N/A

RETURNS: ITEM QUERY

FUNCTIONAL DETAILS: (Pre-defined)

1. Enter ITEM PRINT QUERY.

CALLED BY: VALIDATE AND ECHO PRINT SCREEN

MODULE: GET ROOM

PURPOSE: To get a ROOM, for a given course/lab, entered by the system user.

USES: N/A

RETURNS: ROOM NUMBER
BUILDING CODE

FUNCTIONAL DETAILS: (Pre-defined)

1. Enter BUILDING CODE and ROOM NUMBER.

CALLED BY: VALIDATE AND ECHO COURSE/LAB SCHEDULING SCREEN

MODULE: GET SCHEDULE CHANGES

PURPOSE: To get the schedule information changes for updating schedule records.

USES: N/A

RETURNS: SCHED INFO

FUNCTIONAL DETAILS:

1. Get the schedule information changes for a particular course and its related SCHED RECORDS.

2. Return the schedule info.

CALLED BY: UPDATE SCHEDULE RECORDS

MODULE: GET SCHEDULE RECORD QUERY

PURPOSE: To get a SCHEDULE RECORD QUERY from the system user.

USES: N/A

RETURNS: SCHEDULE RECORD QUERY

FUNCTIONAL DETAILS: (Pre-defined)

1. Enter SCHEDULE RECORD QUERY

CALLED BY: VALIDATE AND ECHO COURSE/LAB SCHEDULING SCREEN

MODULE: GET TIME PERIOD

PURPOSE: To get a TIME PERIOD, for a given course/lab, entered by the system user.

USES: N/A

RETURNS: DAY
TIME

FUNCTIONAL DETAILS: (Pre-defined)

1. Enter DAY(s) and TIME(s).

CALLED BY: VALIDATE AND ECHO COURSE/LAB SCHEDULING SCREEN

MODULE: GET TRANSACCTION

PURPOSE: To get TRANSACTION from system user.

USES: N/A

RETURNS: TRANSACTION

FUNCTIONAL DETAILS: (Pre-defined)

1. Enter TRANSACTION.

CALLED BY: VALIDATE AND ECHO TRANSACTION SCREEN

MODULE: GET USER PASSWORD

PURPOSE: To get the password from the system user.

USES: N/A

RETURNS: USER PASSWORD

FUNCTIONAL DETAILS: (Pre-defined)

1. Enter USER PASSWORD.

CALLED BY: VALIDATE AND ECHO PASSWORD SCREEN

MODULE: GET VALID DATA MAINTENANCE TRANSACTION

PURPOSE: To get a valid data maintenance transaction.

USES: N/A

RETURNS: VALID TRANSACTION

FUNCTIONAL DETAILS:

```
Begin
  Call DISPLAY TRANSACTION SCREEN (DATA MAINTENANCE
                                   TRANSACTIONS)
  Call VALIDATE AND ECHO TRANSACTION SCREEN
  RETURN VALID TRANSACTION
End
```

CALLED BY: SCHEDULING DATA MAINTENANCE

MODULE: GET VALID EVENT/COURSE SCHEDULING TRANSACTION

PURPOSE: To get a valid event/course scheduling transaction.

USES: N/A

RETURNS: VALID TRANSACTION

FUNCTIONAL DETAILS:

```
Begin
  Call DISPLAY TRANSACTION SCREEN (EVENT/COURSE
                                   TRANSACTIONS)
  Call VALIDATE AND ECHO TRANSACTION SCREEN
  Return VALID TRANSACTION
End
```

CALLED BY: MANAGE EVENT/COURSE SCHEDULING

MODULE: GET VALID FILE

PURPOSE: To get a valid file to update.

USES: N/A

RETURNS: VALID FILE

FUNCTIONAL DETAILS:

```
Begin
  Call DISPLAY FILE SCREEN
  Call VALIDATE AND ECHO FILE SCREEN
  RETURN VALID FILE
End
```

CALLED BY: UPDATE DATABASE FILES
REVIEW OLD SCHEDULE FILE
DELETE SCHEDULE FILE

MODULE: GET VALID FILE NAME

PURPOSE: To get a VALID FILE NAME for a newly created file.

USES: N/A

RETURNS: VALID FILE NAME

FUNCTIONAL DETAILS:

```
Begin
  Call DISPLAY FILE SCREEN
  Call VALIDATE AND ECHO FILE SCREEN
  Return VALID FILE NAME
End
```

CALLED BY: SAVE TO MASTER SCHEDULE FILE

MODULE: GET VALID FILE UPDATE TRANSACTION

PURPOSE: To get a valid file update transaction.

USES: N/A

RETURNS: VALID TRANSACTION

FUNCTIONAL DETAILS:

```
Begin
  Call DISPLAY TRANSACTION SCREEN (UPDATE
                                   TRANSACTIONS)
  Call VALIDATE AND ECHO TRANSACTION SCREEN
  Return VALID TRANSACTION
End
```

CALLED BY: UPDATE DATABASE FILE

MODULE: GET VALID ITEM TO PRINT

PURPOSE: To get a valid file, record or report to print.

USES: N/A

RETURNS: VALID ITEM

FUNCTIONAL DETAILS:

```
Begin
  Call DISPLAY PRINT SCREEN
  Call VALIDATE AND ECHO PRINT SCREEN
  Return VALID ITEM
End
```

CALLED BY: PRINT FILE, RECORD, REPORT

MODULE: GET VALID NPS SCHEDULING SYSTEM TRANSACTION

PURPOSE: To get a valid NPS Scheduling System transaction.

USES: N/A

RETURNS: VALID TRANSACTION

FUNCTIONAL DETAILS:

```
Begin
  Call DISPLAY TRANSACTION SCREEN (NPS3
                                   TRANSACTIONS)
  Call VALIDATE AND ECHO TRANSACTION SCREEN
  Return VALID TRANSACTION
End
```

CALLED BY: NPS SCHEDULING SYSTEM

MODULE: GET VALID PASSWORD

PURPOSE: To get a valid password from the user, in order to allow the user to enter the NPS scheduling system.

USES: N/A

RETURNS: ENTER

FUNCTIONAL DETAILS:

```
Begin
  Call DISPLAY PASSWORD SCREEN
  Call VALIDATE AND ECHO PASSWORD SCREEN
  If flag = VALID PASSWORD
    Return ENTER
  End if
End
```

CALLED BY: NPS SCHEDULING SYSTEM

MODULE: GET VALID SCHEDULE RECORD TO UPDATE

PURPOSE: To get the VALID SCHEDULE RECORDS the system user wishes to update.

USES: N/A

RETURNS: VALID SCHEDULE RECORDS

FUNCTIONAL DETAILS:

```
Begin
  Call DISPLAY SCHEDULE RECORD SCREEN
  Call VALIDATE AND ECHO SCHEDULE RECORD SCREEN
  Return VALID SCHEDULE RECORDS
End
```

CALLED BY: UPDATE SCHEDULE FILE

MODULE: GET VALID TIME PERIOD AND ROOM

PURPOSE: To get a VALID TIME PERIOD and VALID ROOM for a given course/course segment and lab.

USES: SCREEN INDICATOR
COURSE SEGMENT SCREEN INFO

RETURNS: VALID TIME PERIOD
VALID ROOM

FUNCTIONAL DETAILS:

```
Begin
  Call DISPLAY COURSE/LAB SCHEDULING SCREEN
    (COURSE SEGMENT SCREEN INFO, SCREEN INDICATOR)

  Call VALIDATE AND ECHO COURSE/LAB SCHEDULING
                                                SCREEN
  Return (VALID TIME PERIOD, VALID ROOM)
End
```

CALLED BY: SCHEDULE REFRESHER COURSES
SCHEDULE COURSES w/ REQUIRED TIME PERIOD
SCHEDULE COURSES w/ REQUIRED TIME PERIOD FOR INSTRUCTOR
SCHEDULE COURSES w/ LARGE # OF STUDENTS
SCHEDULE COURSES TAUGHT BY DEAN/CHAIRMAN
SCHEDULE SIMULTANEOUS SCHEDULED COURSES
SCHEDULE ACCELERATED COURSES
SCHEDULE COURSES W/ DESIGNATED STUDENT COURSE GRPS
SCHEDULE COURSES REQUIRING SPECIAL ROOMS
SCHEDULE COURSES W/ 3 HOUR LABS
SCHEDULE COURSES W/ REQUIRED LAB ROOM AND TIME PERIOD
SCHEDULE COURSES W/ DIVERSITY OF MAJORS
SCHEDULE COURSES W/ TEACHING TEAMS
SCHEDULE REMAINING COURSES

MODULE: MANAGE EVENT/COURSE SCHEDULING

PURPOSE: To manage the scheduling of semi-permanant events and courses.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call GET VALID EVENT/COURSE SCHEDULING
                                        TRANSACTION
    Case VALID TRANSACTION
      When VALID TRANSACTION = CREATE
        Call CREATE NEW SCHEDULES
      When VALID TRANSACTION = REVIEW
        Call REVIEW OLD SCHEDULE FILE
      When VALID TRANSACTION = PRINT
        Call PRINT FILE, RECORD, REPORT
      When VALID TRANSACTION = DELETE
        Call DELETE SCHEDULE FILE
    End Case
  End loop
End
```

CALLED BY: NPS SCHEDULING SYSTEM

MODULE: MODIFY DATA RECORD

PURPOSE: To modify a data record in the given file.

USES: VALID FILE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Modify a record in the VALID FILE.

CALLED BY: UPDATE DATABASE FILES

MODULE: NPS SCHEDULING SYSTEM

PURPOSE: To coordinate the scheduling of NPS events, courses and final exams, and associated schedules, reports, graphs and data maintenance.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  CALL_GET VALID PASSWORD
  If flag = ENTER
    Loop
      Call GET VALID NPS SCHEDULING SYSTEM
           TRANSACTION
        Case VALID TRANSACTION
          When VALID TRANSACTION = DATA MAINTENANCE
            Call SCHEDULING DATA MAINTENANCE
          When VALID TRANSACTION = EVENT/COURSE
                                   SCHEDULING
            Call MANAGE EVENT/COURSE SCHEDULING
          When VALID TRANSACTION = FINAL EXAM
                                   SCHEDULING
            Call MANAGE FINAL EXAM SCHEDULING
          When VALID TRANSACTION = MASTER INSTRUCTION
                                   SCHEDULE
            Call MANAGE MASTER INSTRUCION SCHEDULE
          When VALID TRANSACTION = DATA GRAPH
            Call MANAGE SCHEDULING DATA GRAPH
        End case
    End loop
  End if
End
```

CALLED BY: N/A

MODULE: PRINT

PURPOSE: To print the VALID ITEM.

USES: VALID ITEM

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Print the VALID ITEM.

CALLED BY: PRINT FILE, RECORD, REPORT

MODULE: PRINT FILE, RECORD, REPORT

PURPOSE: To coordinate printing of files, records and reports.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call GET VALID ITEM TO PRINT
    Call PRINT (ITEM)
  End loop
End
```

CALLED BY: SCHEDULING DATA MAINTENANCE
MANAGE EVENT/COURSE SCHEDULING

MODULE: PUT ALTERNATIVE SELECTION TO SCHED RECORDS

PURPOSE: To put the alternative selected to the related SCHED RECORDS.

USES: SCHED RECORDS
UPDATED SCHED INFO

RETURNS: UPDATED SCHED RECORDS

FUNCTIONAL DETAILS: (Pre-defined)

1. For each SCHED RECORD, put appropriate SCHED INFO in the slot where UPDATED SCHED INFO.TIME PERIOD = SCHED RECORD.TIME PERIOD.

CALLED BY: RESOLVE SCHEDULE RECORD CONFLICTS
RESOLVE UPDATE CONFLICTS

MODULE: PUT COURSE/LAB SCHEDULING SCREEN

PURPOSE: To display the COURSE/LAB SCHEDULING SCREEN on the CRT.

USES: COURSE/LAB SCHEDULING SCREEN
INVALID TIME PERIOD MESSAGE
INVALID ROOM MESSAGE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Display the COURSE/LAB SCHEDULING SCREEN on the CRT screen.

2. Display INVALID TIME PERIOD MESSAGE or INVALID ROOM MESSAGE when appropriate.

CALLED BY: DISPLAY COURSE/LAB SCHEDULING SCREEN
VALIDATE AND ECHO COURSE/LAB SCHEDULING SCREEN

MODULE: PUT FILE SCREEN

PURPOSE: To display the FILE SCREEN on the CRT.

USES: FILE SCREEN
INVALID FILE MESSAGE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Display the FILE SCREEN on the CRT screen.
2. DISPLAY INVALID FILE MESSAGE when appropriate.

CALLED BY: DISPLAY FILE SCREEN
VALIDATE AND ECHO FILE SCREEN

MODULE: PUT HELP SCREEN

PURPOSE: To display the help screen on the CRT.

USES: HELP SCREEN

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Display the given HELP SCREEN on the CRT screen.

CALLED BY: DISPLAY HELP SCREEN

MODULE: PUT PASSWORD SCREEN

PURPOSE: To display the password screen on the CRT.

USES: PASSWORD SCREEN
INVALID PASSWORD MESSAGE 1
INVALID PASSWORD MESSAGE 2

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Display the password screen on the CRT screen.
2. DISPLAY INVALID PASSWORD MESSAGE 1 or INVALID PASSWORD MESSAGE 2 when apprpriate.

CALLED BY: DISPLAY PASSWORD SCREEN
VALIDATE AND ECHO PASSWORD SCREEN

MODULE: PUT PRINT SCREEN

PURPOSE: To display the print screen on the CRT.

USES: PRINT SCREEN
INVALID PRINT MESSAGE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Display the PRINT SCREEN on the CRT screen.
2. Display INVALID PRINT MESSAGE when appropriate.

CALLED BY: DISPLAY PRINT SCREEN
VALIDATE AND ECHO PRINT SCREEN

MODULE: PUT SCHEDULE CONFLICT/ALTERNATIVES SCREEN

PURPOSE: To display the SCHEDULE CONFLICT/ALTERNATIVES SCREEN on the CRT.

USES: SCHEDULE CONFLICT/ALTERNATIVES SCREEN
INVALID SELECTION MESSAGE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Display the SCHEDULE CONFLICT/ALTERNATIVES SCREEN on the CRT screen.

2. Display INVALID SELECTION MESSAGE when appropriate.

CALLED BY: DISPLAY SCHEDULE CONFLICT/ALTERNATIVES SCREEN
VALIDATE AND ECHO SCHEDULE CONFLICT/ALTERNATIVES SCREEN

MODULE: PUT SCHEDULE RECORD SCREEN

PURPOSE: To display the SCHEDULE RECORD SCREEN on the CRT.

USES: SCHEDULE RECORD SCREEN
INVALID SCHEDULE RECORD MESSAGE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Display the SCHEDULE RECORD SCREEN on the CRT screen.
2. Display INVALID SCHEDULE RECORD MESSAGE when appropriate.

CALLED BY: DISPLAY SCHEDULE RECORD SCREEN
VALIDATE AND ECHO SCHEDULE RECORD SCREEN

MODULE: PUT TRANSACTION SCREEN

PURPOSE: To display the TRANSACTION SCREEN on the CRT screen.

USES: TRANSACTION SCREEN
INVALID TRANSACTION MESSAGE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Display the TRANSACTION SCREEN on the CRT screen.
2. Display the INVALID TRANSACTION MESSAGE when appropriate.

CALLED BY: DISPLAY TRANSACTION SCREEN
VALIDATE AND ECHO TRANSACTION SCREEN

MODULE: READ COURSE FILE

PURPOSE: To read the COURSE FILE.

USES:
NULL STATUS
REFRESHER COURSE
COURSE REQUIRED TIME
INSTRUC REQUIRED TIME
COURSE TYPE
COURSE NUMBER
SIMULTANEOUS SCHED COURSE
ACCELERATED COURSE
SPECIAL ROOM REQUIREMENT
LAB HRS
REQUIRED LAB ROOM
DIVERSITY OF MAJORS

RETURNS: COURSE INFO

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the next COURSE RECORD, until EOF where the given data matches the COURSE RECORD data and where COURSE RECORD.STATUS is NULL STATUS.

2. Read the next COURSE RECORD, until EOF, where COURSE RECORD.COURSE TYPE and COURSE RECORD.COURSE NUMBER equals the given COURSE TYPE and COURSE NUMBER, respectively.

3. Return the desired COURSE INFO.

CALLED BY:
SCHEDULE REFRESHER COURSES
SCHEDULE COURSES w/ REQUIRED TIME PERIOD
SCHEDULE COURSES w/ REQUIRED TIME PERIOD FOR INSTRUCTOR
SCHEDULE COURSES w/ LARGE # OF STUDENTS
SCHEDULE COURSES TAUGHT BY DEAN/CHAIRMAN
SCHEDULE SIMULTANEOUS SCHEDULED COURSES
SCHEDULE ACCELERATED COURSES
SCHEDULE COURSES W/ DESIGNATED STUDENT COURSE GRPS
SCHEDULE COURSES REQUIRING SPECIAL ROOMS
SCHEDULE COURSES W/ 3 HOUR LABS
SCHEDULE COURSES W/ REQUIRED LAB ROOM AND TIME PERIOD
SCHEDULE COURSES W/ DIVERSITY OF MAJORS
SCHEDULE COURSES W/ TEACHING TEAMS
SCHEDULE REMAINING COURSES

MODULE: READ COURSE SEGMENT FILE

PURPOSE: To read the COURSE SEGMENT FILE.

USES:
COURSE TYPE
COURSE NUMBER
NULL STATUS
SEGMENT STUDENTS
INSTRUC CODE
DESIGNATED STUDENT COURSE GRP
TEACHING TEAM

RETURNS:
COURSE SEGMENT RECORD
EOF

FUNCTIONAL DETAILS: (Pre-defined)

```
Begin
  Read next COURSE SEGMENT RECORD
  For COURSE SEGMENT RECORD.COURSE TYPE = COURSE
        TYPE and COURSE SEGMENT RECORD.COURSE
                    NUMBER = COURSE NUMBER
    Return COURSE SEGMENT RECORD
  For COURSE SEGMENT RECORD.STATUS = NULL STATUS
        and COURSE SEGMENT.SEGMENT STUDENTS >=
                    SEGMENT STUDENTS
    Return COURSE SEGMENT RECORD
  For COURSE SEGMENT RECORD.INSTRUC CODE = INSTRUC
        CODE and COURSE SEGMENT RECORD.STATUS =
                    NULLSTATUS
    Return COURSE SEGMENT RECORD
  For COURSE SEGMENT RECORD.STATUS = NULL STATUS
        and COURSE SEGMENT.DESIGNATED STUDENT
                    COURSE GRP not null
    Return COURSE SEGMENT RECORD
  For COURSE SEGMENT RECORD.STATUS = NULL STATUS
        and COURSE SEGMENT.TEACHING TEAM not null
    Return COURSE SEGMENT RECORD
  For EOF
    Return EOF
End
```

CALLED BY: SCHEDULE REFRESHER COURSES
SCHEDULE COURSES w/ REQUIRED TIME PERIOD
SCHEDULE COURSES w/ REQUIRED TIME PERIOD FOR INSTRUCTOR
SCHEDULE COURSES w/ LARGE # OF STUDENTS
SCHEDULE COURSES TAUGHT BY DEAN/CHAIRMAN
SCHEDULE SIMULTANEOUS SCHEDULED COURSES
SCHEDULE ACCELERATED COURSES
SCHEDULE COURSES W/ DESIGNATED STUDENT COURSE GRPS
SCHEDULE COURSES REQUIRING SPECIAL ROOMS
SCHEDULE COURSES W/ 3 HOUR LABS
SCHEDULE COURSES W/ REQUIRED LAB ROOM AND TIME PERIOD
SCHEDULE COURSES W/ DIVERSITY OF MAJORS
SCHEDULE COURSES W/ TEACHING TEAMS
SCHEDULE REMAINING COURSES

MODULE: READ FILE

PURPOSE: To read the given file.

USES: FILENAME

RETURNS: FILE

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the FILE that matches the FILENAME given.

CALLED BY: VALIDATE AND ECHO FILE SCREEN

MODULE: READ HELP INSTRUCTION FILE

PURPOSE: To read the file containing help instructions for system screen displays.

USES: HELP SCREEN INDICATOR

RETURNS: HELP INSTRUCTIONS

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the file containing the HELP INSTRUCTIONS.
2. RETURN the HELP INSTRUCTIONS for the given HELP SCREEN INDICATOR.

CALLED BY: DETERMINE HELP INSTRUCTIONS

MODULE: READ INSTRUC SCHED FILE

PURPOSE: To read the INSTRUC SCHED FILE and find the related INSTRUC SCHED RECORDS.

USES: INSTRUC CODE

RETURNS: INSTRUC SCHED RECORD
EOF (flag)

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the INSTRUC SCHED FILE and return the related INSTRUC SCHED RECORDS pertaining to the given INSTRUC CODE.

2. Read the Instructor schedule file and return all the INSTRUC SCHED RECORDS for saving to master schedule file.

CALLED BY: DETERMINE MBRSHP SCHEDS AND ROOM SCHED
DETERMINE RELATED SCHEDULE RECORD COMFLICTS
SAVE TO MASTER SCHEDULE FILE

MODULE: READ INSTRUCTOR FILE

PURPOSE: To read the Instructor file.

USES: DEPT CHAIR OR DEAN

RETURNS: FACULTY CODE
DEPT LTR CODE

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the next Instructor record and return the FACULTY CODE and DEPT LTR CODE for the Instructor that has a DEPT CHAIR OR DEAN indicator.

CALLED BY: SCHEDULE COURSES TAUGHT BY DEAN/CHAIRMAN

MODULE: READ M/F E-Z CLASS DEMAND LIST FILE

PURPOSE: To read the mainframe file containing course information.

USES: N/A

RETURNS: COURSE TYPE
COURSE NUMBER
LECT HRS
LAB HRS

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the mainframe file containing course information.

CALLED BY: UPDATE COURSE/COURSE SEGMENT FILES

MODULE: READ M/F PROFESSOR LISTING FILE

PURPOSE: To read the mainframe file containing Faculty information.

USES: N/A

RETURNS: FACULTY NAME
FACULTY CODE
DEPT LTR CODE

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the mainframe file containing Faculty information.

CALLED BY: UPDATE FACULTY FILE

MODULE: READ M/F SCG SECT FILE

PURPOSE: To read the mainframe file containing STUDENT COURSE GRP records.

USES:
COURSE TYPE
COURSE NUMBER

RETURNS:
SECT SCG RECORD
EDF (flag)
SCG CARD NUMBER

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the mainframe file containing STUDENT COURSE GRP records.

CALLED BY:
UPDATE STUDENT COURSE CROUP FILE
UPDATE COURSE/COURSE SEGMENT FILES

MODULE: READ M/F STUDENT SECT FILE

PURPOSE: To read the mainframe file containing student records.

USES: N/A

RETURNS: EOF
SECT STUDENT RECORD

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the mainframe file containing student records.

CALLED BY: UPDATE STUDENT FILE

MODULE: READ ROOM SCHED FILE

PURPOSE: To read the ROOM SCHED FILE and find the related ROOM SCHED RECORD.

USES: ROOM

RETURNS: ROOM SCHED RECORD
EOF (flag)

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the ROOM SCHED FILE and return the related ROOM SCHED RECORD pertaining to the given ROOM.

2. Read the ROOM schedule file and return all ROOM SCHED RECORDS to save to master schedule file.

CALLED BY: DETERMINE MBRSHP SCHEDS AND ROOM SCHED
DETERMINE RELATED SCHEDULE RECORD CONFLICTS
SAVE TO MASTER SCHEDULE FILE

MODULE: READ SCG SCHED FILE

PURPOSE: To read the SCG SCHED FILE and find all related SCG SCHED RECORDS.

USES: DESIGNATED STUDENT COURSE GRP'S

RETURNS: SCG SCHED RECORD
EOF (flag)

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the SCG SCHED FILE and return all the related SCG SCHED RECORDS pertaining to the given DESIGNATED STUDENT COURSE GRP'S.

2. Read the SCG schedule file and return all SCG SCHED RECORDS to be saved to master schedule file.

CALLED BY: DETERMINE MBRSHP SCHEDS AND ROOM SCHED
DETERMINE RELATED SCHEDULE RECORD CONFLICTS
SAVE TO MASTER SCHEDULE FILE

MODULE: READ SCHEDULE RECORDS

PURPOSE: To read the schedule records pertaining to the SCHEDULE RECORD QUERY.

USES: SCHEDULE RECORD QUERY

RETURNS: SCHED RECORDS
NOT FOUND (flag)

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the SCHED RECORDS that pertain to the given SCHEDULE RECORD QUERY.

CALLED BY: VALIDATE AND ECHO SCHEDULE RECORD SCREEN

MODULE: READ SCREEN

PURPOSE: To read the screen currently displayed on the CRT.

USES: N/A

RETURNS: FLAG TYPE (flag)
PRINT SCREEN
PASSWORD SCREEN
COURSE/LAB SCHEDULING SCREEN
PASSWORD HELP SCREEN
TRANSACTION SCREEN
FILE SCREEN
SCHEDULE CONFLICT/ALTERNATIVES SCREEN
SCHEDULE RECORD SCREEN

FUNCTIONAL DETAILS: (Pre-defined)

1. READ the screen and return user designated FLAG TYPE entry.

CALLED BY: VALIDATE AND ECHO PASSWORD SCREEN
DETERMINE HELP INSTRUCTIONS
VALIDATE AND ECHO TRANSACTION SCREEN
VALIDATE AND ECHO FILE SCREEN
VALIDATE AND ECHO PRINT SCREEN
VALIDATE AND ECHO COURSE/LAB SCHEDULING SCREEN
VALIDATE AND ECHO SCHEDULE CONFLICT/ALTERNATIVES SCREEN
VALIDATE AND ECHO SCHEDULE RECORD SCREEN

MODULE: READ SEMI-PERM EVENT FILE

PURPOSE: To read the SEMI-PERM EVENT FILE.

USES: N/A

RETURNS: EOF (flag)
SEMI-PERM EVENT RECORD

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the SEMI-PERM EVENT FILE and return SEMI-PERM EVENT RECORD each time called until EOF is reached.

CALLED BY: DETERMINE SEMI-PERM EVENT AND EVENT MBRSHP

MODULE: READ SEMI-PERM EVENT MBRSHP FILE

PURPOSE: To read the SEMI-PERM EVENT MBRSHP FILE

USES: EVENT TYPE
EVENT NAME

RETURNS: MBRSHP

FUNCTIONAL DETAILS: (Pre-defined)

1. Read the SEMI-PERM EVENT MBSHP FILE and return all the MBRSHP RECORDS that relate to the given event.

CALLED BY: DETERMINE SEMI-PERM EVENT AND EVENT MBRSHP

MODULE: READ SYSTEM PASSWORD FILE

PURPOSE: To read the system password.

USES: N/A

RETURNS: SYSTEM PASSWORD

FUNCTIONAL DETAILS: (Pre-defined)

1.Read the file containing the system password.

CALLED BY: VALIDATE AND ECHO PASSWORD SCREEN

MODULE: RESOLVE SCHED RECORD CONFLICT

PURPOSE: To allow the system user to resolve TIME PERIOD SCHEDULE CONFLICTS.

USES: SCHED INFO

RETURNS: UPDATED SCHED RECORDS
UPDATED SCHED INFO

FUNCTIONAL DETAILS:

```
Begin
  Call DETERMINE RELATED SCHEDULE RECORD CONFLICTS
                                          (SCHED INFO)
  If NO CONFLICT (flag) is returned
    Return (UPDATED SCHED RECORDS, SCHED INFO as
                                  UPDATED SCHED INFO)
  Else
    Call SEARCH for ALTERNATIVES (SCHED RECORDS)
    Call DISPLAY SCHEDULE CONFLICTS/ALTERNATIVES
                              SCREEN (ALTERNATIVES,
                                SCHEDULE CONFLICTS)
    Call VALIDATE AND ECHO SCHEDULE
                          CONFLICT/ALTERNATIVESSCREEN
    SCHED INFO.TIME PERIOD := VALID ALTERNATIVE
                                          SELECTION
    UPDATED SCHED INFO := SCHED INFO
    Call PUT ALTERNATIVE SELECTION TO SCHEDULE
                              RECORDS (UPDATED SCHED
                                INFO, SCHED RECORDS)
    Return (UPDATED SCHED RECORDS, UPDATED SCHED
                                          INFO)
  End if
End
```

CALLED BY: SCHEDULE REFRESHER COURSES
SCHEDULE COURSES w/ REQUIRED TIME PERIOD
SCHEDULE COURSES w/ REQUIRED TIME PERIOD FOR INSTRUCTOR
SCHEDULE COURSES w/ LARGE # OF STUDENTS
SCHEDULE COURSES TAUGHT BY DEAN/CHAIRMAN
SCHEDULE SIMULTANEOUS SCHEDULED COURSES
SCHEDULE ACCELERATED COURSES
SCHEDULE COURSES W/ DESIGNATED STUDENT COURSE GRPS
SCHEDULE COURSES REQUIRING SPECIAL ROOMS
SCHEDULE COURSES W/ 3 HOUR LABS
SCHEDULE COURSES W/ REQUIRED LAB ROOM AND TIME PERIOD
SCHEDULE COURSES W/ DIVERSITY OF MAJORS
SCHEDULE COURSES W/ TEACHING TEAMS
SCHEDULE REMAINING COURSES

MODULE: RESOLVE UPDATE CONFLICTS

PURPOSE: To resolve the conflicts pertaining to updates made to SCHED RECORDS.

USES: SCHED INFO
SCHEDULE CONFLICT
SCHED RECORDS

RETURNS: UPDATED SCHED RECORDS

FUNCTIONAL DETAILS:

```
Begin
  Call SEARCH FOR ALTERNATIVES (SCHED RECORDS)
  Call DISPLAY SCHEDULE/ALTERNATIVES SCREEN
                  (ALTERNATIVES, SCHEDULE CONFLICT)
  Call VALIDATE AND ECHO SCHEDULE
                       CONFLICT/ALTERNATIVES SCREEN
  SCHED INFO.TIME PERIOD := VALID ALTERNATIVE
                                          SELECTION
  UPDATED SCHED INFO := SCHED INFO
  Call PUT ALTERNATIVE SELECTION TO SCHEDULE
      RECORDS (UPDATED SCHED INFO, SCHED RECORDS)
  Return (UPDATED SCHED RECORDS)
End
```

CALLED BY: UPDATE SCHEDULE RECORDS

MODULE: SAVE TO MASTER SCHEDULE FILE

PURPOSE: To coordinate saving created schedule records to a master schedule file.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call GET VALID FILE NAME
  Call VALIDATE AND ECHO FILE SCREEN
  Call READ INSTRUC SCHED FILE
  Call READ ROOM SCHED FILE
  Call READ STUDENT COURSE GRP SCHED FILE
  Call WRITE MASTER SCHEDULE FILE (VALID FILE NAME,
                                   SCHED RECORDS)
End
```

CALLED BY: CREATE NEW SCHEDULES

MODULE: SAVE TO PERTINENT FILE/RECORD

PURPOSE: To save the updated records and information to the affected files and records.

USES: UPDATED SCHED RECORDS
VALID TIME PERIOD
NO OF COURSE SEGMENTS

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  STATUS := DONE
  If UPDATED SCHED INFO.SEGMENT NUMBER = NO OF
                                    COURSE SEGMENTS
    Call UPDATE COURSE RECORD (STATUS, COURSE
                              TYPE, COURSE NUMBER)
  Endif
  Call UPDATE COURSE SEGMENT RECORD (STATUS,
                                VALID TIME PERIOD,
                      COURSE TYPE, COURSE NUMBER)
  Call GET DATE
  Call WRITE STUDENT COURSE GRP SCHED FILE (DATE,
                    UPDATED STUDENT COURSE GRP SCHED
                                              RECORD)
  Call WRITE ROOM SCHED FILE (DATE, UPDATED ROOM
                                          SCHEDRECORD)
  Call WRITE INSTRUC SCHED FILE (DATE, UPDATED
                                  INSTRUC SCHED RECORD)
End
```

CALLED BY: SCHEDULE REFRESHER COURSES
SCHEDULE COURSES w/ REQUIRED TIME PERIOD
SCHEDULE COURSES w/ REQUIRED TIME PERIOD FOR INSTRUCTOR
SCHEDULE COURSES w/ LARGE # OF STUDENTS
SCHEDULE COURSES TAUGHT BY DEAN/CHAIRMAN
SCHEDULE SIMULTANEOUS SCHEDULED COURSES
SCHEDULE ACCELERATED COURSES
SCHEDULE COURSES W/ DESIGNATED STUDENT COURSE GRPS
SCHEDULE COURSES REQUIRING SPECIAL ROOMS
SCHEDULE COURSES W/ 3 HOUR LABS
SCHEDULE COURSES W/ REQUIRED LAB ROOM AND TIME PERIOD
SCHEDULE COURSES W/ DIVERSITY OF MAJORS
SCHEDULE COURSES W/ TEACHING TEAMS
SCHEDULE REMAINING COURSES

MODULE: SCHEDULE ACCELERATED COURSES

PURPOSE: To coordinate scheduling of courses that are accelerated courses.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ COURSE FILE (NULL STATUS, ACCELERATED
                                                COURSE)
    Loop until Course Segment EOF
      Call READ COURSE SEGMENT FILE (COURSE TYPE,
                                          COURSE NUMBER)
      Call GET VALID TIME PERIOD AND ROOM (COURSE
             SEGMENT SCREEN INFO, SCREEN INDICATOR)
      Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                                INFO)
      Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                  SCHED RECORDS, VALID TIME PERIOD,
                       NO OF SEGMENTS, COURSE TYPE,
                                          COURSE NUMBER)
    End loop
  End loop
End
```

CALLED BY: SCHEDULE MIDDLE PRIORITY COURSES

MODULE: SCHEDULE COURSES REQUIRING SPECIAL ROOMS

PURPOSE: To coordinate scheduling courses requiring special rooms for instruction.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ COURSE FILE (NULL STATUS, SPECIAL
                                    ROOM REQUIREMENT)
    Loop until Course Segment EOF
      Call READ COURSE SEGMENT FILE (COURSE TYPE,
                                    COURSE NUMBER)
      Call GET VALID TIME PERIOD AND ROOM (COURSE
            SEGMENT SCREEN INFO, SCREEN INDICATOR)
      Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                    INFO)
      Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                  SCHED RECORDS, VALID TIME PERIOD,
                       NO OF SEGMENTS, COURSE TYPE,
                                    COURSE NUMBER)
    End loop
  End loop
End
```

CALLED BY: SCHEDULE MIDDLE PRIORITY COURSES

MODULE: SCHEDULE COURSES TAUGHT BY DEAN/CHAIRMAN

PURPOSE: To coordinate scheduling courses taught by a Dean or Department Chairman.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ INSTRUCTOR FILE (DEPT CHAIR OR DEAN)
    Loop until Course Segment EOF
      Call READ COURSE SEGMENT FILE (NULL STATUS,
                                     INSTRUC CODE)
      Call READ COURSE FILE (COURSE TYPE, COURSE
                                     NUMBER)
      Call GET VALID TIME PERIOD AND ROOM (COURSE
            SEGMENT SCREEN INFO, SCREEN INDICATOR)
      Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                     INFO)
      Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                  SCHED RECORDS, VALID TIME PERIOD,
                       NO OF SEGMENTS, COURSE TYPE,
                                     COURSE NUMBER)
  End loop
  If COURSE RECORD.SIMULTANEOUS SCHED COURSE not
                                     null
    Call UPDATE SIMULTANEOUS SCHEDULED COURSE
            (SIMULTANEOUS SCHED COURSE, VALID ROOM,
                                     VALID TIME PERIOD)
  Endif
End loop
End
```

CALLED BY: SCHEDULE MIDDLE PRIORITY COURSES

MODULE: SCHEDULE COURSES W/ 3 HOUR LABS

PURPOSE: To coordinate scheduling courses that have 3 hour laboratories.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ COURSE FILE (NULL STATUS, LAB HRS>/3)
    Loop until Course Segment EOF
      Call READ COURSE SEGMENT FILE (COURSE TYPE,
                                     COURSE NUMBER)
      Call GET VALID TIME PERIOD AND ROOM (COURSE
             SEGMENT SCREEN INFO, SCREEN INDICATOR)
      Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                              INFO)
      Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                  SCHED RECORDS, VALID TIME PERIOD,
                       NO OF SEGMENTS, COURSE TYPE,
                                     COURSE NUMBER)
    End loop
  End loop
End
```

CALLED BY: SCHEDULE LOWER PRIORITY COURSES

MODULE: SCHEDULE COURSES W/ DESIGNATED STUDENT COURSE GRPS

PURPOSE: To coordinate scheduling of courses that have designated Student Course Groups.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until COURSE SEGMENT EOF
    Call READ COURSE SEGMENT FILE (NULL STATUS,
                        DESIGNATED STUDENT COURSE GRP)
    Call READ COURSE FILE (COURSE TYPE, COURSE
                                              NUMBER)
    Call GET VALID TIME PERIOD AND ROOM (COURSE
              SEGMENT SCREEN INFO, SCREEN INDICATOR)
    Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                              INFO)
    Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                    SCHED RECORDS, VALID TIME PERIOD,
                         NO OF SEGMENTS, COURSE TYPE,
                                    COURSE NUMBER)
  End loop
End
```

CALLED BY: SCHEDULE MIDDLE PRIORITY COURSES

MODULE: SCHEDULE COURSES W/ DIVERSITY OF MAJORS

PURPOSE: To coordinate scheduling of courses that have a diversity of majors in its enrollment.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ COURSE FILE (NULL STATUS, DIVERSITY
                                          OF MAJORS)
    Loop until Course Segment EOF
      Call READ COURSE SEGMENT FILE (COURSE TYPE,
                                     COURSE NUMBER)
      Call GET VALID TIME PERIOD AND ROOM (COURSE
             SEGMENT SCREEN INFO, SCREEN INDICATOR)
      Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                              INFO)
      Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                  SCHED RECORDS, VALID TIME PERIOD,
                       NO OF SEGMENTS, COURSE TYPE,
                                     COURSE NUMBER)
    End loop
  End loop
End
```

CALLED BY: SCHEDULE LOWER PRIORITY COURSES

MODULE: SCHEDULE COURSES W/ LARGE # OF STUDENTS

PURPOSE: To coordinate scheduling courses that have a large enrollment.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ COURSE SEGMENT FILE (NULL STATUS,
                                   SEGMENT STUDENTS)
    Call READ COURSE FILE (COURSE TYPE, COURSE
                                        NUMBER)
    Call GET VALID TIME PERIOD AND ROOM (COURSE
         SEGMENT SCREEN INFO, SCREEN INDICATOR)
    Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                            INFO)
    Call SAVE TO PERTINENT FILE/RECORD (UPDATED
         SCHED RECORDS, VALID TIME PERIOD,
              NO OF SEGMENTS, COURSE TYPE,
                             COURSE NUMBER)
  End loop
  If COURSE RECORD.SIMULTANEOUS SCHED COURSE not
                                          null
    Call UPDATE SIMULTANEOUS SCHEDULED COURSE
        (SIMULTANEOUS SCHED COURSE, VALID ROOM,
                        VALID TIME PERIOD)
  Endif
 End loop
End
```

CALLED BY: SCHEDULE TOP PRIORITY COURSES

MODULE: SCHEDULE COURSES W/ REQUIRED LAB ROOM AND TIME PERIOD

PURPOSE: To coordinate scheduling courses that require a specific laboratory at a specific time.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ COURSE FILE (NULL STATUS, REQUIRED
                                              LABROOM)
    Loop until Course Segment EOF
      Call READ COURSE SEGMENT FILE (COURSE TYPE,
                                        COURSE NUMBER)
      Call GET VALID TIME PERIOD AND ROOM (COURSE
             SEGMENT SCREEN INFO, SCREEN INDICATOR)
      Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                              INFO)
      Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                  SCHED RECORDS, VALID TIME PERIOD,
                       NO OF SEGMENTS, COURSE TYPE,
                                        COURSE NUMBER)
    End loop
  End loop
End
```

CALLED BY: SCHEDULE LOWER PRIORITY COURSES

MODULE: SCHEDULE COURSES w/REQUIRED TIME PERIOD

PURPOSE: To schedule courses with required times of instruction.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
     Call READ COURSE FILE (NULL STATUS, COURSE
                                    REQUIRED TIME)
     Loop until EOF
       Call READ COURSE SEGMENT FILE (COURSE TYPE,
                                    COURSENUMBER)
       Call GET VALID TIME PERIOD AND ROOM (COURSE
            SEGMENT SCREEN INFO, SCREEN INDICATOR)
       Call RESOLVE SCHED RECORD CONFLICTS (SCHED
                                    INFO)
       Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                         SCHED RECORDS, COURSE TYPE,
                  COURSE NUMBER, VALID TIME PERIOD,
                                    NO OF SEGMENTS)
     Endloop
     If COURSE RECORD.SIMULTANEOUS SCHED COURSE
                                          notnull
     Call UPDATE SIMULTANEOUS SCHEDULED COURSE
                         (SIMULTANEOUS SCHED COURSE,
                VALID ROOM, VALID TIME PERIOD)
     End if
   Endloop
 End
```

CALLED BY: SCHEDULE TOP PRIORITY COURSES

MODULE: SCHEDULE COURSES w/REQUIRED TIME PERIOD FOR INSTRUCTOR

PURPOSE: To schedule courses that require Instructor to teach at specific TIME PERIODS.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ COURSE FILE (NULL STATUS, INSTRUC
                                        REQUIRED TIME)
    Loop until COURSE SEGMENT EOF
      Call READ COURSE SEGMENT FILE (COURSE TYPE,
                                        COURSENUMBER)
      Call GET VALID TIME PERIOD AND ROOM (COURSE
            SEGMENT SCREEN INFO, SCREEN INDICATOR)
      Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                              INFO)
      Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                          SCHED RECORDS, COURSE TYPE,
                  COURSE NUMBER, VALID TIME PERIOD,
                                        NO OF SEGMENTS)
    End loop
    If COURSE RECORD SIMULTANEOUS SCHED COURSE no
                                                null
      Call UPDATE SIMULTANEOUS SCHEDULED COURSE
                (SIMULTANEOUS SCHED COURSE, VALID
                          ROOM, VALID TIME PERIOD)
    Endif
  Endloop
End
```

CALLED BY: SCHEDULE TOP PRIORITY COURSES

MODULE: SCHEDULE COURSES W/ TEACHING TEAM

PURPOSE: To coordinate scheduling courses that are taught by teaching teams.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until COURSE SEGMENT EOF
    Call READ COURSE SEGMENT FILE (NULL STATUS,
                                   TEACHING TEAM)
    Call READ COURSE FILE (COURSE TYPE, COURSE
                                        NUMBER)
    Call GET VALID TIME PERIOD AND ROOM (COURSE
              SEGMENT SCREEN INFO, SCREEN INDICATOR)
    Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                            INFO)
    Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                  SCHED RECORDS, VALID TIME PERIOD,
                       NO OF SEGMENTS, COURSE TYPE,
                                   COURSE NUMBER)
  End loop
End
```

CALLED BY: SCHEDULE LOWER PRIORITY COURSES

MODULE: SCHEDULE LOWER PRIORITY COURSES

PURPOSE: To coordinate the scheduling of lower priority courses.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call SCHEDULE COURSES W/ 3 HOUR LABS
  Call SCHEDULE COURSES W/ REQUIRED LAB ROOM AND
                                        TIME PERIOD
  Call SCHEDULE COURSES W/ DIVERSITY OF MAJORS
  Call SCHEDULE COURSES W/ TEACHING TEAM
  Call SCHEDULE REMAINING COURSES
End
```

CALLED BY: CREATE NEW SCHEDULES

MODULE: SCHEDULE MIDDLE PRIORITY COURSES

PURPOSE: To coordinate the scheduling of middle priority courses.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call SCHEDULE COURSES TAUGHT BY DEAN/CHAIRMAN
  Call SCHEDULE SIMULTANEOUS SCHEDULED COURSES
  Call SCHEDULE ACCELERATED COURSES
  Call SCHEDULE COURSES W/ DESIGNATED STUDENT
                                        COURSE GRPS
  Call SCHEDULE COURSES REQUIRING SPECIAL ROOMS
End
```

CALLED BY: CREATE NEW SCHEDULES

MODULE: SCHEDULE REFRESHER COURSES

PURPOSE: To schedule Refresher Courses.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop
     Call READ COURSE FILE (NULL STATUS, REFRESHER
                                                COURSE)
     Loop until EOF
       Call READ COURSE SEGMENT FILE (COURSE TYPE,
                                        COURSE NUMBER)
       Call GET VALID TIME PERIOD AND ROOM (COURSE
            SEGMENT SCREEN INFO, SCREEN INDICATOR)
       Call RESOLVE SCHEDULE RECORD CONFLICTS
                                          (SCHED INFO)
       Call SAVE TO PERINTENT FILE/RECORD (UPDATED
                          SCHED RECORDS, COURSE TYPE,
                  COURSE NUMBER, VALID TIME PERIOD,
                                      NO_OF_SEGMENTS)
     Endloop
     If COURSE RECORD.SIMULTANEOUS SCHED COURSE
                                              not null
     Call UPDATE SIMULTANEOUS SCHEDULED COURSE
            (SIMULTANEOUS SCHED COURSE, VALID ROOM,
                                  VALID TIME PERIOD)
     End if
   End loop
 End
```

CALLED BY: SCHEDULE TOP PRIORITY COURSES

MODULE: SCHEDULE REMAINING COURSES

PURPOSE: To coordinate the scheduling of courses that have yet to be scheduled for the academic quarter.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ COURSE FILE (NULL STATUS)
    Loop until Course Segment EOF
      Call READ COURSE SEGMENT FILE (COURSE TYPE,
                                     COURSE NUMBER)
      Call GET VALID TIME PERIOD AND ROOM (COURSE
             SEGMENT SCREEN INFO, SCREEN INDICATOR)
      Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                              INFO)
      Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                  SCHED RECORDS, VALID TIME PERIOD,
                       NO OF SEGMENTS, COURSE TYPE,
                                     COURSE NUMBER)
    End loop
  End loop
End
```

CALLED BY: SCHEDULE LOWER PRIORITY COURSES

MODULE: SCHEDULE SEMI-PERM EVENTS

PURPOSE: To automatically schedule semi-permanent events.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call DETERMINE SEMI-PERM EVENT AND EVENT MBRSHP
    Call DETERRMINE MBRSHP SCHED AND ROOM SCHED
                                        (MBRSHP, ROOM)

    Call ASSIGN SEMI-PERM EVENT TO SCHEDULES (SCHED
                    RECORDS, SEMI-PERM EVENT RECORD)

  Endloop
End
```

CALLED BY: CREATE NEW SCHEDULE FILE

MODULE: SCHEDULE SIMULTANEOUS SCHEDULED COURSES

PURPOSE: To coordinate scheduling of courses that must be scheduled simultaneously with other courses.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ COURSE FILE (NULL STATUS,
                                    SIMULTANEOUS COURSE)
    Loop until COURSE SEGMENT EOF
      Call READ COURSE SEGMENT FILE (COURSE TYPE,
                                        COURSE NUMBER)
      Call GET VALID TIME PERIOD AND ROOM (COURSE
             SEGMENT SCREEN INFO, SCREEN INDICATOR)
      Call RESOLVE SCHEDULE RECORD CONFLICTS (SCHED
                                                INFO)
      Call SAVE TO PERTINENT FILE/RECORD (UPDATED
                            SCHED RECORDS, VALID TIME
                              PERIOD, NO OF SEGMENTS,
                          COURSE TYPE, COURSE NUMBER)
    End loop
    If COURSE RECORD.SIMULTANEOUS SCHED COURSE not
                                                null
      Call UPDATE SIMULTANEOUS SCHEDULED COURSE
            (SIMULTANEOUS SCHED COURSE, VALID ROOM,
                                    VALID TIME PERIOD)
    Endif
  End loop
End
```

CALLED BY: SCHEDULE MIDDLE PRIORITY COURSES

MODULE: SCHEDULE TOP PRIORITY COURSES

PURPOSE: To coordinate the scheduling of top priority courses.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call SCHEDULE REFRESHER COURSES
  Call SCHEDULE COURSES w/REQUIRED TIME PERIOD
  Call SCHEDULE COURSES w/LARGE ATTENDANCE
  Call SCHEDULE COURSES TAUGHT BY DEPT CHAIR/DEAN
End
```

CALLED BY: CREATE NEW SCHEDULES

MODULE: SCHEDULING DATA MAINTENANCE

PURPOSE: To coordinate the maintenance of data files for the NPS SCHEDULING SYSTEM.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call GET VALID DATA MAINTENANCE TRANSACTION
      Case VALID TRANSACTION
        When VALID TRANSACTION = COPY FILES
          Call COPY MAINFRAME FILES
        When VALID TRANSACTION = UPDATE
          Call UPDATE DATA FILES
        When VALID TRANSACTION = PRINT
          Call PRINT FILE, RECORD, REPORT
      End Case
  End loop
End
```

CALLED BY: NPS SCHEDULING SYSTEM

MODULE: SEARCH DATABASE

PURPOSE: To search the database for an ITEM based on the ITEM QUERY.

USES: ITEM QUERY

RETURNS: ITEM

FUNCTIONAL DETAILS: (Pre-defined)

1. Search database for the ITEM matching the ITEM QUERY.

CALLED BY: VALIDATE AND ECHO PRINT SCREEN

MODULE: SEARCH FOR ALTERNATIVES

PURPOSE: To search the given SCHED RECORDS for an alternative TIME PERIOD slot that has no conflict.

USES: SCHED RECORDS

RETURNS: ALTERNATIVES

FUNCTIONAL DETAILS: (Pre-defined)

1. Search the given SCHED RECORDS and find all the alternative TIME PERIOD slots, to schedule the course, where NO CONFLICT exists between the given SCHED RECORDS.

2. Return all TIME PERIODs where INSTRUC SCHED RECORD.TIME PERIOD + ROOM SCHED RECORD.TIME PERIOD + STUDENT COURSE GRP.TIME PERIOD = null.

CALLED BY: RESOLVE SCHEDULE RECORD CONFLICTS
RESOLVE UPDATE CONFLICTS

MODULE: UPDATE COURSE/COURSE SEGMENT FILES

PURPOSE: To update the COURSE/COURSE SEGMENT FILES with data from the mainframe E-Z Class Demand List File.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ M/F E-Z CLASS DEMAND LIST FILE
    Call READ M/F SCG SECT FILE (COURSE TYPE,
                                 COURSE NUMBER)
    Call WRITE COURSE FILE (COURSE TYPE, COURSE
       NUMBER, LECT HRS, LAB HRS, SCG CARD NUMBER)
    Call WRITE COURSE SEGMENT FILE (COURSE TYPE,
                                    COURSENUMBER)
  End loop
End
```

CALLED BY: COPY MAINFRAME FILES

MODULE: UPDATE COURSE RECORD

PURPOSE: To update the status of a given course.

USES: STATUS
COURSE TYPE
COURSE NUMBER

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. For the given COURSE TYPE and COURSE NUMBER, update COURSE RECORD.STATUS to the given STATUS.

CALLED BY: SAVE TO PERTINENT FILE/RECORD

MODULE: UPDATE COURSE SEGMENT RECORD

PURPOSE: To UPDATE the STATUS, DAY and PERIOD of a given COURSE SEGMENT.

USES: STATUS
TIME PERIOD
COURSE TYPE
COURSE NUMBER

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. For the given COURSE TYPE and COURSE NUMBER, update COURSE SEGMENT RECORD.STATUS with STATUS, COURSE SEGMENT RECORD.DAY with DAY and COURSE SEGMENT RECORD.PERIOD with PERIOD.

CALLED BY: SAVE TO PERTINENT FILE/RECORD

MODULE: UPDATE DATABASE FILES

PURPOSE: To coordinate the updating of NPS^3 database files.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call GET VALID FILE
    Call GET VALID UPDATE TRANSACTION
    Case VALID TRANSACTION
      When VALID TRANSACTION = ADD
         Call ADD DATA RECORD
      When VALID TRANSACTION = MODIFY
         Call MODIFY DATA RECORD
      When VALID TRANSACTION = DELETE
         Call DELETE DATA RECORD
    End Case
  End loop
End
```

CALLED BY: SCHEDULING DATA MAINTENANCE

MODULE: UPDATE FACULTY FILE

PURPOSE: To update the Faculty File with data from the mainframe Professor Listing File.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ M/F PROFESSOR LISTING FILE
    Call WRITE FACULTY FILE (FACULTY NAME, DEPT
                             LTR CODE, FACULTY CODE)
  End loop
End
```

CALLED BY: COPY MAINFRAME FILES

MODULE: UPDATE SCHEDULE FILE

PURPOSE: To coordinate updating an old master schedule file.

USES: VALID FILE

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call GET VALID SCHEDULE RECORDS TO UPDATE
    Call UPDATE SCHEDULE RECORDS (VALID SCHED
                                  RECORDS)
  End loop
End
```

CALLED BY: REVIEW OLD SCHEDULE FILE

MODULE: UPDATE SCHEDULE RECORDS

PURPOSE: To update schedule records.

USES: VALID SCHED RECORDS

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Call GET DATE
  Call GET SCHEDULE CHANGES
  Call DETERMINE CONFLICTS (SCHED INFO, SCHED
                                        RECORDS)
  If SCHEDULE CONFLICTS returned
    Call RESOLVE UPDATE CONFLICTS (SCHED INFO,
                SCHEDULE CONFLICT, SCHED RECORDS)
    Call WRITE TO FILE (UPDATED SCHED RECORDS)
  Elseif UPDATED SCHED RECORDS returned
    Call WRITE TO FILE (UPDATED SCHED RECORDS)
  End if
End
```

CALLED BY: UPDATE SCHEDULE FILE

MODULE: UPDATE SIMULTANEOUS SCHEDULED COURSE RECORD

PURPOSE: To update a simultaneously scheduled course with the VALID ROOM and VALID TIME PERIOD.

USES: SIMULTANEOUS SCHEDULED COURSE
VALID ROOM
VALID TIME PERIOD

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. For the given simultaneous scheduled COURSE TYPE and COURSE NUMBER, update COURSE SEGMENT RECORD.DAY with VALID DAY, COURSE SEGMENT RECORD.PERIOD with VALID PERIOD, COURSE SEGMENT RECORD.BUILDING CODE with VALID BUILDING CODE and COURSE SEGMENT RECORD.ROOM NUMBER with VALID ROOM NUMBER.

2. Change COURSE SEGMENT RECORD.STATUS and COURSE RECORD.STATUS for the given simultaneous scheduled COURSE TYPE and COURSE NUMBER to DONE.

CALLED BY: SCHEDULE REFRESHER COURSES
SCHEDULE COURSES w/ REQUIRED TIME PERIOD
SCHEDULE COURSES w/ REQUIRED TIME PERIOD FOR INSTRUCTOR
SCHEDULE COURSES w/ LARGE NUMBER OF STUDENTS
SCHEDULE COURSES TAUGHT BY DEAN/CHAIRMAN
SCHEDULE SIMULTANEOUS SCHEDULED COURSES

MODULE: UPDATE STUDENT COURSE GRP FILE

PURPOSE: To update the STUDENT COURSE GRP FILE with data from the mainframe SECT file.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ M/F SCG SECT FILE
    Call WRITE SCG SCHED FILE (SECT SCG RECORD)
  End loop
End
```

CALLED BY: COPY MAINFRAME FILES

MODULE: UPDATE STUDENT FILE

PURPOSE: To update the STUDENT FILE with data from the mainframe SECT file.

USES: N/A

RETURNS: N/A

FUNCTIONAL DETAILS:

```
Begin
  Loop until EOF
    Call READ M/F STUDENT SECT FILE
    Call WRITE STUDENT FILE (SECT STUDENT
                                    RECORDS)
  End loop
End
```

CALLED BY: COPY MAINFRAME FILES

MODULE: VALIDATE AND ECHO COURSE/LAB SCHEDULING SCREEN

PURPOSE: To validate the time period and room entered by the system user for a given course/lab.

USES: N/A

RETURNS: VALID TIME PERIOD
VALID ROOM

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call READ SCREEN
    If FLAG TYPE = QUIT
      Exit system
    Elseif FLAG TYPE=HELP
     . Call DETERMINE HELP INSTRUCTIONS (COURSE/LAB
                                    SCHEDULING SCREEN)

    Elseif FLAG TYPE = RETURN
      GO TO previous menu
    End if
    Call GET TIME PERIOD
    Call GET ROOM
    If TIME PERIOD and ROOM are valid
      Return (VALID TIME PERIOD, VALID ROOM)
    Else Call PUT COURSE/LAB SCHEDULING SCREEN
      (COURSE/LAB SCHEDULING SCREEN, INVALID TIME
      PERIOD MESSAGE AND/OR INVALID ROOM MESSAGE)
    End if
  End Loop
End
```

CALLED BY: GET VALID TIME PERIOD AND ROOM

MODULE: VALIDATE AND ECHO FILE SCREEN

PURPOSE: To get a VALID FILE to update.

USES: N/A

RETURNS: VALID FILE
VALID FILE NAME

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call READ SCREEN
    If FLAG TYPE = HELP
      Call DETERMINE HELP INSTRUCTIONS (FILE
                                        SCREEN)
    Elseif FLAG TYPE = QUIT
      Exit system
    Elseif FLAG TYPE = RETURN
      GO TO previous menu
    Endif
    Call GET FILE NAME
    When updating, reviewing or deleting
      Call READ FILE (FILE NAME)
      If FILENAME Exists as a FILE
        Return (VALID FILE)
      Else Call Put FILE SCREEN (FILE SCREEN,
                                 INVALID FILE MESSAGE)
      Endif
    When creating file
      Return (VALID FILE NAME)
      Exit loop
  End loop
End
```

CALLED BY: GET VALID FILE

MODULE: VALIDATE AND ECHO PASSWORD SCREEN

PURPOSE: To get a valid system password.

USES: N/A

RETURNS: VALID PASSWORD (flag)

FUNCTIONAL DETAILS:

```
Begin
  Count := 1
  Loop while Count 1 = 3
    Call READ SCREEN
    If FLAGTYPE = HELP
      Call Determine HELP INSTRUCTIONS (PASSWORD
                                        SCREEN)
    Elseif FLAG TYPE = QUIT
      EXIT SYSTEM
    End if
    Call GET USER PASSWORD
    Count = Count + 1
    Call READ SYSTEM PASSWORD
    If SYSTEM PASSWORD = USER PASSWORD
      Return VALID PASSWORD
    Else Call Put PASSWORD SCREEN (PASSWORD
              SCREEN, INVALID PASSWORD MESSAGE 1)
    Endif
  End loop
  Call PUT PASSWORD SCREEN (PASSWORD SCREEN,
                    INVALID PASSWORD MESSAGE 2)
  EXIT SYSTEM
End
```

CALLED BY: GET VALID PASSWORD

MODULE: VALIDATE AND ECHO PRINT SCREEN

PURPOSE: To get a valid item to print

USES: N/A

RETURNS: VALID ITEM

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call READ SCREEN
    IF FLAG TYPE = HELP
      Call DETERMINE HELP INSTRUCTIONS (PRINT
                                        SCREEN)
    Elseif FLAG TYPE = QUIT
      Exit system
    Elseif FLAG TYPE = RETURN
      GO TO previous menu
    Endif
    Call GET ITEM PRINT QUERY
    Call SEARCH DATABASE (ITEM QUERY)
    If ITEM exists for ITEM QUERY
      Return VALID ITEM
    Else Call PUT FILE SCREEN (FILE SCREEN,
                               INVALID FILE MESSAGE)
    Endif
  End loop
End
```

CALLED BY: GET VALID ITEM TO PRINT

MODULE: VALIDATE AND ECHO SCHEDULE CONFLICT/ALTERNATIVES SCREEN

PURPOSE: To validate the alternative TIME PERIOD selected by the system user for a given SCHEDULE CONFLICT.

USES: N/A

RETURNS: VALID ALTERNATIVE SELECTION

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call READ SCREEN
    If FLAG TYPE = QUIT
       Exit system
    Elseif FLAG TYPE = HELP
        Call DETERMINE HELP INSTRUCTIONS (SCHEDULE
                                CONFLICT/ALTERNATIVES
                                              SCREEN)
    Elseif FLAG TYPE = RETURN
        GO TO previous menu
    Endif
    Call GET ALTERNATIVE SELECTION
    If ALTERNATIVE SELECTION is VALID
      Return (VALID ALTERNATIVE SELECTION)
    Else Call PUT SCHEDULE CONLICT/ALTERNATIVES
            SCREEN (SCHEDULE CONFLICT/ALTERNATIVES
                SCREEN, INVALID SELECTION MESSAGE)
    Endif
  Endloop
End
```

CALLED BY: RESOLVE SCHEDULE RECORD CONFLICT
RESOLVE UPDATE CONFLICTS

MODULE: VALIDATE AND ECHO SCHEDULE RECORD SCREEN

PURPOSE: To coordinate getting VALID SCHED RECORDS to update.

USES: N/A

RETURNS: VALID SCHED RECORDS

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call READ SCREEN
    IF FLAG TYPE = HELP
      Call DETERMINE HELP INSTRUCTIONS (SCHEDULE
                                        RECORD SCREEN)
    Elseif FLAG TYPE = QUIT
      Exit system
    Elseif FLAG TYPE = RETURN
      GO TO previous menu
    Endif
    Call GET SCHEDULE RECORD QUERY
    Call READ SCHED RECORDS (SCHEDULE RECORD QUERY)
    If SCHED RECORDS exist
      Return VALID SCHED RECORDS
    Elseif NOT FOUND returned
      Call PUT SCHEDULE RECORD SCREEN (SCHEDULE
                                       RECORD SCREEN,
                        INVALID SCHEDULE RECORD MESSAGE)
    End if
  End loop
End
```

CALLED BY: GET VALID SCHEDULE RECORDS TO UPDATE

MODULE: VALIDATE AND ECHO TRANSACTION SCREEN

PURPOSE: To get a valid transaction.

USES: N/A

RETURNS: VALID TRANSACTION

FUNCTIONAL DETAILS:

```
Begin
  Loop
    Call READ SCREEN
    If FLAG TYPE = HELP
      Call DETERMINE HELP INSTRUCTIONS (TRANSACTION
                                        SCREEN)
    Elseif FLAG TYPE = QUIT
      Exit system
    Elseif FLAG TYPE = RETURN
      GO TO previous menu
    Endif
    Call GET TRANSACTION
    If TRANSACTION is a VALID TRANSACTION
      Return VALID TRANSACTION
    Else Call PUT TRANSACTION SCREEN (TRANSACTION
              SCREEN, INVALID TRANSACTION MESSAGE)
    Endif
  End loop
End
```

CALLED BY: GET VALID NPS SCHEDULING SYSTEM TRANSACTION
GET VALID DATA MAINTENANCE TRANSACTION
GET VALID FILE UPDATE TRANSACTION
GET VALID EVENT/COURSE TRANSACTION

MODULE: WRITE COURSE FILE

PURPOSE: To write a Course file for the NPS^3 from the E-Z Class Demand List File and the SCG SECT File.

USES:
COURSE TYPE
COURSE NUMBER
LECT HRS
LAB HRS
SCG CARD NUMBER

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Write the course information to Course File.

CALLED BY: UPDATE COURSE/COURSE SEGMENT FILES

MODULE: WRITE COURSE SEGMENT FILE

PURPOSE: To write a Course Segment File for the NPS^3 from the E-Z Class Demand List File.

USES: COURSE TYPE
COURSE NUMBER

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Write the course information to Course Segment File.

CALLED BY: UPDATE COURSE/COURSE SEGMENT FILES

MODULE: WRITE FACULTY FILE

PURPOSE: To write a FACULTY FILE for the NPS^3 from the Professor Listing File.

USES: FACULTY NAME
FACULTY CODE
DEPT LTR CODE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Write the Faculty information to FACULTY FILE.

CALLED BY: UPDATE FACULTY FILE

MODULE: WRITE INSTRUC SCHED FILE

PURPOSE: To write the UPDATED INSTRUC SCHED RECORD to the INSTRUC SCHED FILE.

USES: UPDATED INSTRUC SCHED RECORD
DATE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Write the UPDATED INSTRUC SCHED RECORD to the INSTRUC SCHED FILE.

CALLED BY: ASSIGN SEMI-PERM EVENT TO SCHEDULES
SAVE TO PERTINENT FILE/RECORD

MODULE: WRITE MASTER SCHEDULE FILE

PURPOSE: To write the SCHED RECORDS to a master schedule file.

USES: VALID FILE NAME
SCHED RECORDS

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Write all SCHED RECORDS to the designated named file.

CALLED BY: SAVE TO MASTER SCHEDULE FILE

MODULE: WRITE NPS^3 SCG SCHED FILE

PURPOSE: To write a SCG SCHED FILE for the NPS^3 from SECT SCG RECORDS.

USES: SECT SCG RECORD

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Write the SECT SCG RECORD to the NPS^3 SCG SCHED FILE.

CALLED BY: UPDATE STUDENT COURSE GROUP FILE

MODULE: WRITE ROOM SCHED FILE

PURPOSE: To write the UPDATED ROOM SCHED RECORD to the ROOM SCHED FILE.

USES: UPDATED ROOM SCHED RECORD
DATE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Write the UPDATED ROOM SCHED RECORD to the ROOM SCHED FILE.

CALLED BY: ASSIGN SEMI-PERM EVENT TO SCHEDULES
SAVE TO PERTINENT FILE/RECORD

MODULE: WRITE SCG SCHED FILE

PURPOSE: To write the UPDATED SCG SCHED RECORD to the SCG SCHED FILE.

USES: UPDATED SCG SCHED RECORD
DATE

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Write the UPDATED SCG SCHED RECORD to the SCG SCHED FILE.

CALLED BY: ASSIGN SEMI-PERM EVENT TO SCHEDULES
SAVE TO PERTINENT FILE/RECORD

MODULE: WRITE STUDENT FILE

PURPOSE: To write a STUDENT FILE for the NPS^3 from SECT STUDENT RECORDS.

USES: SECT STUDENT RECORDS

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Write the SECT RECORD to the STUDENT FILE.

CALLED BY: UPDATE STUDENT FILE

MODULE: WRITE TO FILE

PURPOSE: To write the UPDATED SCHED RECORDS to the file being updated.

USES: UPDATED SCHED RECORDS

RETURNS: N/A

FUNCTIONAL DETAILS: (Pre-defined)

1. Write the UPDATED SCHED RECORDS to the master schedule file being updated.

CALLED BY: UPDATE SCHEDULE RECORDS

LIST OF REFERENCES

Anderson, Ron, "High-priced, high-end Omnis 7 brings DBMS development to Windows," *PC Magazine*, v11, n9, p. 54, 12 May 1992.

Blyth Software Inc., *Omnis 5 Programmer's Reference*, 1990.

Fogel, Steve, "Double Duty Database," *LAN Magazine*, v7, n5, pp. 149-152, May 1992.

Gardner, Karen, "Designing knowledge systems with objects," *AI Expert*, v6, n9, pp. 32-39, September 1991.

Hass, M.S. and Hochstetler, M.L., *Information Engineering of the Curricular Officer's Segment of a Unified Student Academic Database System for NPS*, Master's Thesis, Naval postgraduate School, Monterey, California, September 1991.

Interview between Edith Phillips & Lisa Quidelig, Naval Postgraduate School Schedulers, and the author, 13 April 1992.

Keyes, Jessica, "How software is developed undergoing basic changes; with GUI's, servers, objects and parallelism, the question remains: How best to write software?," *Software Magazine*, v12, n1, pp. 38-47, January 1992.

Kroenke, D. M. and Dolan, K. A., *Database Processing: Fundamentals, Design, Implementation*, pp. 131-209, Third Edition, Science Research Associates, Inc., 1988.

Keuffel, Warren, "Controlled Rapid Prototyping," *Databased Advisor*, v10, n3, pp. 152-157, March 1992.

Kueffel, Warren, "Faking top-down development," *Computer Language*, v8, n9, pp.35-40, September 1991.

Kueffel, Warren, "Transformation strategies," *Computer Language*, v8, n10, pp.35-43, October 1991.

Miley, Michael, "Database strategies: relational databases," *MacUser*, v8, n6, pp. 60-71, June 1992.

Nolan, Jefferey S. and Youngblood, Phillip D., *Naval Postgraduate School Scheduling Support System*, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1992.

Page-Jones, M., *The Practical Guide to Structured Systems Design*, pp. 1-235, Yourdon Press, 1988.

PC User, "Multi-user databases," n187, pp. 120-141, 17 June 1992.

Plauger, P. J., "Remedial software engineering," *Computer Language*, v9, n3, pp. 23-28, March 1992.

Ragan, Mathew C., *The Administrative Sciences Department Travel Database System*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1992.

Riehle, Richard, "Object lessons: what we mean when we talk about software engineering," *HP Professional*, v5, n11, pp. 76-79, November 1991.

The Computer Conference Analysis Newsletter, "A new generation," n298, p. 6, 15 April 1992.

Wetherbe, James C., *Systems Analysis and Design: Traditional, Structured, and Advanced Concepts and Techniques*, Second Edition, p. 157, West Publishing Company, 1984.

Whitten, J. L., Bentley, L. B. and Barlow, V. M., *Systems Analysis & Design Methods*, pp. 2-427, Second Edition, Irwin, 1989.

INITIAL DISTIBUTION LIST

1. Defense Technical Information Center — 2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 52 — 2
   Naval Postgraduate School
   Monterey, California 93943-5000

3. Mrs. Edith Phillips, Code 612 — 1
   Scheduler
   Naval Postgraduate School
   Monterey, California 93943-5000

4. Mr. Tracy Hammond, Code 61 — 1
   Registrar
   Naval Postgraduate School
   Monterey, California 93943-5000

5. Prof. Barry Frew, Code 05 — 1
   Dean of Computer and Information Services
   Naval Postgraduate School
   Monterey, California 93943-5000

6. Prof. Daniel Dolk, Code AS/Dk — 1
   Department of Administrative Sciences
   Naval Postgraduate School
   Monterey, California 93943-5000

7. Prof. David Erickson, Code CS/Er — 1
   Department of Computer Science
   Naval Postgraduate School
   Monterey, California 93943-5000

8. Computer Technology Curriculum Officer, Code 37 — 1
   Naval Postgraduate School
   Monterey, California 93943-5000

9. Mr. Michael Troian, Code 611 — 1
   Management Analyst
   Naval Postgraduate School
   Monterey, California 93943-5000

10. Mr. Lloyd Nolan — 1
    Senior Programming Analyst
    Naval Postgraduate School
    Monterey, California 93943-5000

11. Commandant of the Marine Corps 1
Code TE 06
Headquarters, U.S. Marine Corps
Washington, D.C. 20380-0001

12. Captain Richard C. Dowler, USMC 1
Headquarters Marine Corps (MCC 008)
HQBN, HQMC, Henderson Hall
Arlington, VA 22214

Thesis
D7018 Dowler
c.1 Naval Postgraduate
School Scheduling System
(NPS^3).

DUDLEY KNOX LIBRARY
3 2768 00034088 9